1-1-1993

Standard For Information Technology-protocols For Distributed Interactive Simulation Applications: Proposed IEEE Standard Draft, Version 2.0 Third Draft

University of Central Florida Institute for Simulation and Training

INSTITUTE FOR SIMULATION AND TRAINING

Contract Number N61339-91-C-0091
STRICOM
DMSO
May 28, 1993

Proposed IEEE Standard Draft STANDARD for INFORMATION TECHNOLOGY - PROTOCOLS for DISTRIBUTED INTERACTIVE SIMULATION APPLICATIONS

VERSION 2.0
Third Draft

INSTITUTE FOR SIMULATION & TRAINING

IST-CR-93-15

INSTITUTE FOR SIMULATION AND TRAINING

Contract Number N61339-91-C-0091
STRICOM
DMSO
May 28, 1993

Proposed IEEE Standard Draft

Standard for Information Technology - Protocols for Distributed Interactive Simulation Applications

Version 2.0
Third Draft

Institute for Simulation and Training
12424 Research Parkway, Suite 300
Orlando FL 32826

University of Central Florida
Division of Sponsored Research

IST-CR-93-15

DRAFT STANDARD
28 MAY 1993

STANDARD FOR INFORMATION TECHNOLOGY

PROTOCOLS FOR

DISTRIBUTED INTERACTIVE SIMULATION APPLICATIONS

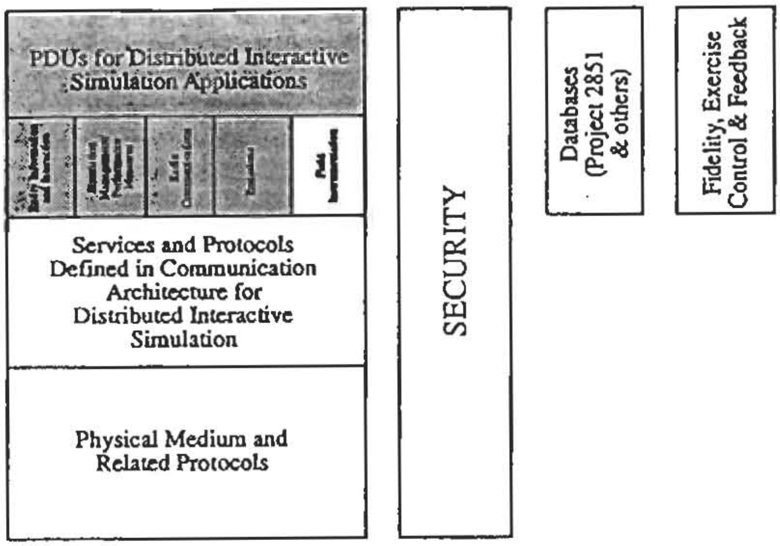
VERSION 2.0

(THIRD DRAFT)

"NOTE: This draft, dated 28 May 1993, prepared by the Institute for Simulation and Training for STRICOM, has not been approved and is subject to modification. DO NOT USE PRIOR TO APPROVAL."

Foreword

This standard is part of a set of standards for Distributed Interactive Simulation (DIS). The relationship between this standard and other DIS standards is shown in the figure below.



0447-5159

Documentation Relationships

This set of standards deals with requirements for simulations participating in a Distributed Interactive Simulation. There are several elements that make up the DIS environment. Each element is addressed by one or more standard documents. Used together, these standards will define an interoperable simulated battle environment.

The main elements addressed by these standards are:

(1) Communications
(2) Simulation Environment
(3) Fidelity, Exercise Control, and Feedback Requirements

The scope of this document lies within the first element, Communications. Its purpose is to define the data messages that are exchanged between simulation applications. These Protocol Data Units (PDUs) provide data concerning simulated entity states and the types of entity interactions that take place in a DIS exercise. Future versions of this standard will contain additional PDUs required to exchange information about interactions and functions not currently supported. The rationale behind the content of this document is contained in the "Rationale Document - Protocols for Distributed Interactive Simulation Applications", IST-CR-93-03.

A related draft standard, "Communication Architecture for Distributed Interactive Simulation" (CADIS) IST-CR-93-13 establishes the requirements for the communication architecture to be used in DIS applications. It makes recommendations concerning the communication profiles that can provide the services to meet those requirements. The standard described by this document, along with the CADIS document, provide the necessary information exchange for the communications element of DIS.

In the second element, Simulation Environment, the government's Project 2851 is providing a military standard describing database formats for terrain, culture, and dynamic model representation. The draft military standard "Standard Simulator Data Base (SSDB) Interchange Format (SIF) for High Detail Input/Output (SIF/HDI) and Distributed Processing (SIF/DP)" is recommended for use with the developing DIS standards.

The required fidelity correlation between simulations in a DIS exercise is addressed in the draft standard "Fidelity Description Requirements for Distributed Interactive Simulation", IST-CR-93-04. The proposed method for setup and control of a DIS exercise and providing feedback at the end is addressed in the draft standard "Exercise Control and Feedback Requirement," IST-CR-93-05.

The area covered by this standard is undergoing evolution. Revisions are anticipated to this standard within the next few years to clarify existing material, to correct possible errors, and to incorporate new related material.

CONTENT

LIST OF FIGURES

LIST OF FIGURES IN ANNEX A

LIST OF ENUMERATION TABLES IN SECTION 6

1. Introduction

1.1 Terminology

1.1.1 Distributed Interactive Simulation (DIS). The DIS concept may be summarized as follows:

DIS is a time and space coherent synthetic representation of world environments designed for linking the interactive, free play activities of people in operational exercises. The synthetic environment is created through real-time exchange of data units between distributed, computationally autonomous simulation applications in the form of simulations, simulators, and instrumented equipment interconnected through standard computer communicative services. The computational simulation entities may be present in one location or may be distributed geographically.

DIS concepts are described in further detail in 1.2.

1.1.2 Host Computer. A computer that supports one or more simulation applications. All host computers participating in a simulation exercise are connected by a common network.

1.1.3 Simulation Application. A simulation application is the executing software on a host computer that generates one or more simulation entities. The simulation application represents or "simulates" real-world phenomena for the purpose of training or experimentation. Examples of simulation applications include manned vehicle simulators, computer generated forces, and computer interfaces between a DIS network and real equipment. The simulation application receives and processes information concerning entities created by peer simulation applications through the exchange of DIS PDUs. More than one simulation application may simultaneously execute on a host computer. The simulation application is the application layer protocol entity that implements the protocol defined in this document. This document will use the term "simulation application" to avoid confusion between protocol entities and simulation entities. The term "simulation" may also be used in place of simulation application.

1.1.4 Simulation Entity. A simulation entity is an element of the synthetic environment that is created and controlled by a simulation application through the exchange of DIS PDUs. Examples of types of simulated entities are: tank, submarine, carrier, fighter aircraft, missiles, bridges, or other elements of the synthetic environment. It is possible that a simulation application may be controlling more than one simulation entity. Simulation entities may also be referred to as "entities" in this document.

1.1.5 Simulation Exercise. A simulation exercise consists of one or more interacting simulation applications. Simulations participating in the same simulation exercise share a common identifying number called the Exercise Identifier. These simulations also utilize correlated representations of the synthetic environment in which they operate.

1.1.6 Simulation Environment. A simulation environment consists of the operational environment surrounding the simulation entities. This environment includes terrain, atmospheric, and bathospheric information. Although environment information is outside the scope of this standard, it is assumed that participants in the same DIS exercise will be using environment information that is adequately correlated for the type of exercise to be performed.

1.2 Key Concepts. This section contains details on key DIS concepts that will help the reader understand the context of this standard. This information is of a general or explanatory nature that may be helpful, but is not mandatory.

1.2.1 Basic Architecture Concepts. The basic architecture concepts of DIS are an extension of the Simulator Networking (SIMNET) program developed by Defense Advanced Research Project Agency (DARPA). The basic architecture concepts for DIS are:

(1) No central computer controls the entire simulation exercise
(2) Autonomous simulation applications are responsible for maintaining the state of one or more simulation entities
(3) A standard protocol is used for communicating "ground truth" data
(4) Changes in the state of an entity are communicated by simulation applications
(5) Perception of events or other entities is determined by the receiving application
(6) Dead reckoning algorithms are used to reduce communications processing

The implications of each of these concepts as they apply to DIS are discussed in the following paragraphs.

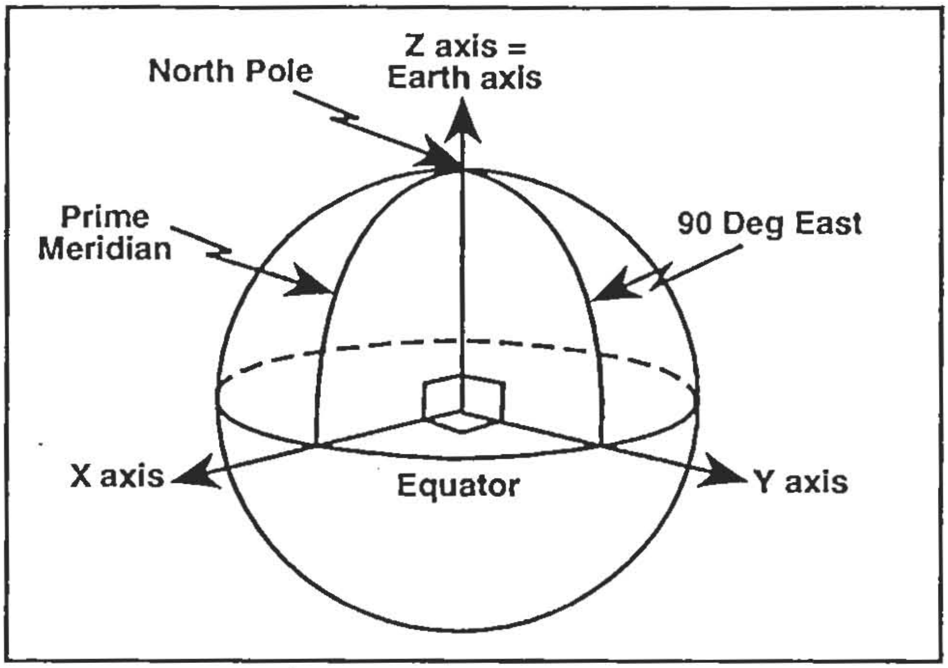
1.2.1.1 No Central Computer. Some war games have a central computer that maintains the world state and calculates the effects of each entity's actions on other entities and the environment. These computer systems must be sized with resources to handle the worst case load for a maximum number of simulated entities. DIS uses a distributed simulation approach in which the responsibility for simulating the state of each entity rests with separate simulation applications residing in host computers connected via a network. As new host computers are added to the network, each new host computer brings its own resources.

1.2.1.2 Autonomous Simulation Applications. Simulation applications (or simulations) are autonomous and generally responsible for maintaining the state of one entity. In some cases, a simulation will be responsible for maintaining the state of several entities. As the user operates controls in the simulated or actual equipment, the simulation is responsible for modeling the resulting actions of the entity using a high fidelity simulation model. That simulation is responsible for sending messages to others, as necessary, to inform them of any observable actions. All simulations are responsible for interpreting and responding to messages of interest from other simulations and maintaining a model of the state of entities represented in the simulation exercise. Simulations may also maintain a model of the state of the environment and nondynamic entities, such as bridges and buildings, that may be intact or destroyed.

1.2.1.3 Ground Truth Versus Perception. Each simulation application communicates the state of the entity it controls (location, orientation, velocity, articulated parts position, etc.) to other simulations on the network. The receiving simulation is responsible for taking this ground truth data and calculating whether the entity represented by the sending simulation is detectable by visual or electronic means. This perceived state of the entity is then displayed to the user as required by the individual simulation.

1.2.1.4 Dead Reckoning. A method of position/ orientation estimation, called dead reckoning, is used to limit the rate at which simulations must issue state updates for an entity. Each simulation maintains a high fidelity model of the entity it represents. In addition, the simulation maintains a simpler model of its entity. The simpler model represents the view of that entity by other simulation applications on the network and is an extrapolation of position and orientation state using a specified dead reckoning algorithm. On a regular basis, the simulation compares the high fidelity model of its entity to the simpler model of the entity. If the difference between the two exceeds a predetermined threshold, the simulation will update the simpler model using the information from the high fidelity model. At the same time, the simulation will send updated information to other simulations on the network so that they can update their model of the entity. By using dead reckoning, simulations are not required to report the status of their entities every frame.

1.2.2 Coordinate Systems. Locations in the simulated world are identified using a right-handed geocentric Cartesian coordinate system called the World Coordinate System. The shape of the world is described by the WGS 84 standard, DMA TR 8350.2. The origin of the coordinate system is the centroid of the earth. The axes of this system are labeled X, Y, and Z, with the positive X-axis passing through the Prime Meridian at the Equator, with the positive Y-axis passing through 90 degrees East longitude at the Equator and the positive Z-axis passing through the North pole as shown in Figure 1-1. A distance of one unit measured in world coordinates corresponds to a distance of one meter in the simulated world, and a straight line in the world coordinate system is a straight line in the simulated world.



0458-5201

Fig 1-1
Geocentric Cartesian Coordinates

To describe the location and orientation of any particular entity, an entity coordinate system is associated with the entity. This is also a right-handed Cartesian coordinate system with the distance of one unit corresponding to one meter as in the world coordinate system. The origin of the entity coordinate system is the center of the entity's bounding volume. The axes are labeled x, y, and z with the positive x-axis pointing to the front of the entity, the positive y-axis pointing to the right side of the entity, and positive z-axis pointing down as shown in Figure 1-2.

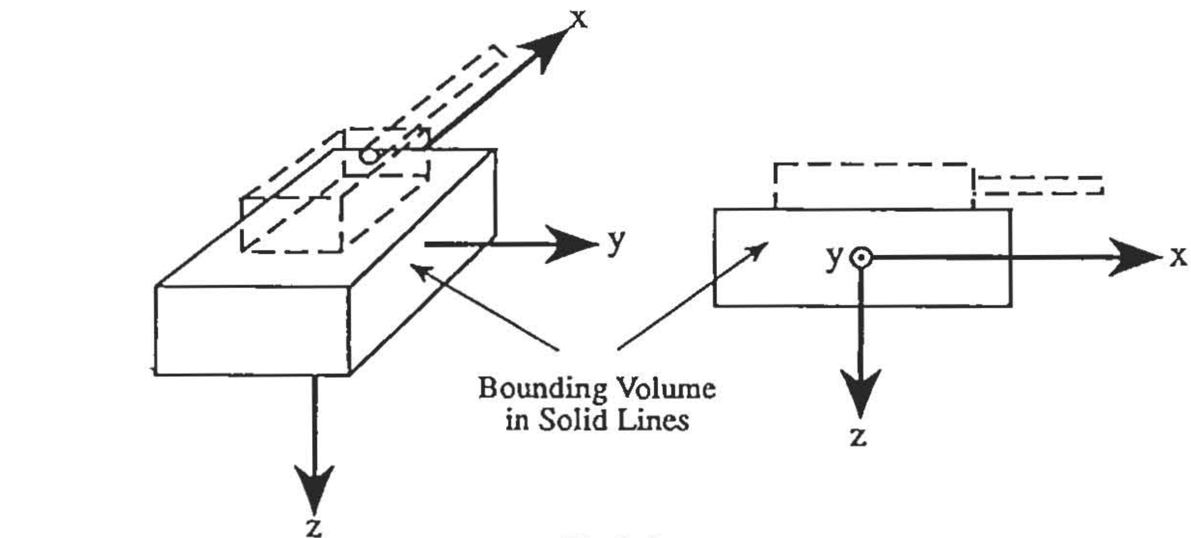
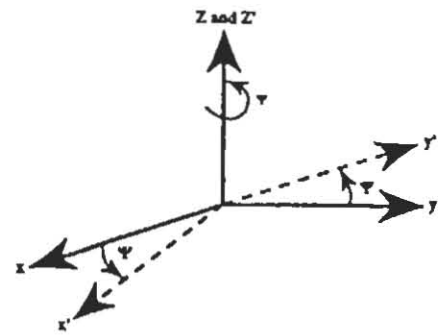
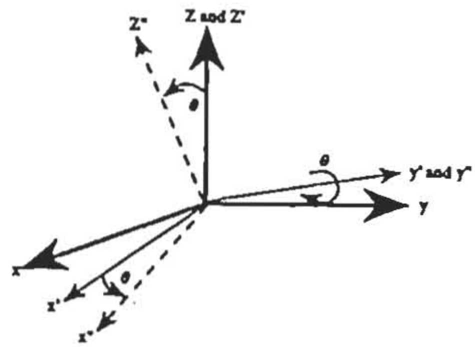


Fig 1-2
Entity Coordinate System

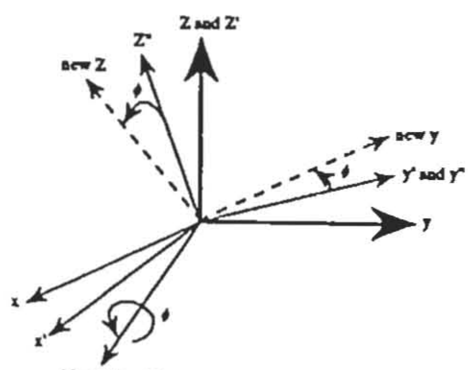
The location of an entity is specified as the position of the origin of its entity coordinate system, expressed in world coordinates. The entity's orientation is specified using three angles that describe the successive rotations needed to transform from the world coordinate system to the entity coordinate system. These angles are called Euler Angles and specify a set of three successive rotations about three different orthogonal axes as shown in Figure 1-3. The order of rotation is as follows: First, rotate about z by the angle psi, then about the new y (y') by angle theta, then about the newest x (x'') by the angle phi. The positive direction of rotation about an axis is defined as clockwise when viewed toward the positive direction along the axis of rotation.



(a)
First rotate about z by angle psi



(b)
Second, rotate about new y (y') by theta



(c)
Third, rotate about newest x (x") by phi

Fig. 1-3
Definition of Euler Angles

1.2.3 Communication Services. The communication services required by each DIS PDU are described in detail in document IST-CR-93-07. A brief summary of the basic communication services necessary for DIS follows:

(1) Data transfer - Each simulation application must be able to transfer data to another simulation application on the network in a single operation, with or without first establishing a logical connection with the destination computer. The unit of data passed in a single operation is called a packet.
(2) Delivery - The communication architecture must support either broadcast, multicast, or unicast packets. Broadcast packets are delivered to all computers on a network. Multicast packets are delivered to a subset of all computers on a network (Broadcasting is actually a special case of multicast). Unicast packets are delivered to a single computer on a network.
(3) Best effort service - The communication architecture should support best effort delivery. Although DIS simulation applications will tolerate occasional failures of the network to deliver packets, these should be allowed to occur only rarely.
(4) Packet Integrity - Transmission errors associated with the network should be detectable by the communication protocol. Corrupted packets should not be delivered to the simulation application.
(5) Performance Requirements - The communication architecture should provide a certain level of performance characterized in terms of throughput and delay. Both network delay and network delay variance should be minimized.

1.2.4 Functional Requirements for DIS. DIS is intended to support the following functional requirements:

(1) Entity information
(2) Entity interaction
(3) Management
(4) Environment information

Currently only the first two requirements are supported by this standard. Enhancement of the first two areas and the addition of capabilities for areas (3) and (4) are the subject of future standards. A brief description of each is given in the paragraphs that follow.

1.2.4.1 Entity Information. The entity information exchanged between simulation applications includes the type of entity, its location, its orientation, and how the entity might appear to others.

(1) Types - The simulation entity could be a vehicle, a building, a munition (such as a missile) or a cloud of

smoke. DIS requires that entities be classified based on their entity type, allowing a variety of different entities to be represented.

(2) Location and Orientation - Sending the location and orientation of an entity is critical for correct representation of the entity by other simulations on the network. Inclusion of the velocity and the acceleration parameters allows receiving simulations to employ higher-level, higher-accuracy extrapolation routines. See 1.2.1.4 on Dead Reckoning.

(3) Appearances - The appearance of an entity can be expressed in a number of ways. Entities may be on fire or smoking. An entity may emit engine smoke or have a wake trailing behind in the water.

1.2.4.2 Entity Interaction. Throughout a simulation exercise, the state information associated with the interactions that take place between entities needs to be exchanged. Interactions that are currently supported include weapons fire, logistics support, and collisions.

(1) Weapons Fire - When an entity fires a weapon, the simulation application controlling the entity needs to communicate the location of the firing weapon and the type of munition fired. The detonation of the munition is also communicated by the simulation application controlling the firing entity. Using the information in the detonation message, all simulation applications controlling affected entities assess damage to their entities.

(2) Logistics Support - Certain services may be modeled in a simulation exercise such as resupply or repair of vehicles. These services are provided under logistics support.

(3) Collisions - In the event that two entities collide, the simulations controlling the entities must be informed of the collision. A message about the collision is sent by each simulation when it detects that its entity has collided with another entity. Each simulation determines the damage to its own entity based on information in the collision message.

1.2.4.3 Future Capabilities. A number of capabilities that are part of this standard are expected to be in future versions or related standards documents. These are described in the following paragraphs.

1.2.4.3.1 Entity Information: Other Types of Appearances. In addition to visual appearance, the entities may also have a certain infrared signature. Also, an entity makes a variety of sounds, some of which may identify the entity under some circumstances. Consequently, each entity has several "appearances," which include emission of sound, light, and heat as well as the reflection of light, radar, and sound (sonar).

1.2.4.3.2 Entity Interaction: Electromagnetic Interactions. The development of technology in the area of sensory data has produced a variety of sensors and emitters, such as sonar, radar, and communications. Representations of these devices are essential in a simulation exercise.

1.2.4.3.3 Management. Centralized management of a simulation exercise is desirable to facilitate the operation of the network and certain aspects of the simulation exercise. DIS management functions can be divided into Network Management and Simulation Management.

(1) Network Management - Network management functions handle the basic network functions such as load management, monitoring of nodes and gateways, and error recovery. The network manager would also have knowledge of host computers on the network, such as their physical locations and network addresses. Analysis of network performance would be performed here.

(2) Simulation Management - There is also a need for centralized management of the simulation exercise. Functions of simulation management include: Start, Restart, Maintenance, and Shutdown of the exercise. Other functions required include the introduction of entities after exercise start, and collection and distribution of certain types of data.

1.2.4.3.4 Environment Information. For simulation entities to participate in the same exercise, they must have access to the same environment information. Different types of information about the environment are necessary to make the exercise as realistic as possible. This information may include changes in the terrain, weather, and ambient illumination.

(1) Changes in the Terrain - Changes in the terrain can be caused by a number of factors. These include engineering effects such as the construction of a bridge or a building; weapons effects that could destroy objects created via engineering effects, as well as change the shape of the terrain due to the impact of shells or explosion of mines; or natural effects such as flooding.

(2) Weather Conditions - Conditions such as rain, snow, fog, or clouds need to be represented in DIS to add to its realism. The wind and its effect on a cloud of smoke can affect vehicle visibility. Chemical clouds and their dispersal can affect dismounted infantry.

(3) Degrees of Ambient Illumination - Night battles, as well as day battles, should be simulated. DIS will support battles that can take place in any degree of ambient illumination.

(4) Other Environmental Effects - The subsurface environment plays a significant role in sub-surface simulations. The effects of water temperature, pressure, and salinity on the propagation of sound will be considered.

1.2.4.3.5 Field Instrumentation. Requirements for real entities on instrumented ranges operating within the DIS environment and interacting with simulated and real entities will be addressed in future versions of this standard. Limited communication bandwidth, volume of data, methods of data transmission, and type of data transmitted are among the issues to be addressed.

1.2.4.3.6 Data Collection and Analysis. DIS may be used to train individuals to work as a team or to evaluate the performance of developmental hardware and software, proposed weapon system platforms, tactics, or operational procedures in a realistic operational scenario. In any case, some data on the performance of exercise participants will be required beyond the data that can be observed from the responses transmitted over the communication medium. This data will be used by the exercise manager to determine the extent to which exercise participants (humans and platforms) achieved the goals of the exercise. Example data includes engine temperature, target detection, target classification, expendables state, and instructor comments.

1.2.4.3.7 Mechanisms for Efficient Use of Network Resources. It may not be necessary that all participants receive all PDUs of an exercise. Mechanisms to allow for efficient use of network resources may include filtering PDUs at various locations in the network, data compression schemes, and sending only changes in the PDU information rather than the entire PDU. Multicast addressing schemes and protocols will be used to create and control groups where PDUs are only sent to members of that group. The multicast mechanisms are currently being developed and will be specified in the areas of communication architecture, network management, and simulation management.

2. General References

The following documents are referenced in this Standard:

(1) DMA TR 8350.2 - Department of Defense World Geodetic System 1984 (WGS 84), Its Definition and Relationships with Local Geodetic Systems. Defense Mapping Agency Technical Report 8350.2, 1987

This document is available from:

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161
(703) 487-4650
Order #: ADA188815

(2) IEEE 754-1985 - IEEE Standard for Binary Floating Point Arithmetic, IEEE Product #SH10116

This is available from:

IEEE Inc.
445 Hoes Lane
P.O. Box 1331
Piscataway, N.J. 08855-1331
USA
Telephone:1-800-678-IEEE

(3) ISO 7498 - Information Processing Systems - Open Systems Interconnection - Basic Reference Model

This is available from:

ISO Central Secretariat
1 rue de Varembe
Case Postale 56
CH-1211
Geneve 20
Switzerland/Suisse

and from:

American National Standards Institute (ANSI)
Sales Department
11 West 42nd Street

New York, NY 10036
USA
Telephone: 212-642-4900

(4) IST-CR-93-02 - Enumeration and Bit Encoded Values for Use with Protocols for Distributed Interactive Simulation Applications (March/93)

(5) IST-CR-93-04 - Draft, Fidelity Description Requirements for Distributed Interactive Simulation

(6) IST-CR-93-05 - Proposed IEEE Draft, Exercise Control and Feedback Requirement for Distributed Interactive Simulation

(7) IST-CR-93-13 - Proposed IEEE Final Draft, Communication Architecture for Distributed Interactive Simulations (CADIS)

All IST documents can be obtained from:

University of Central Florida
Center for Continuing Education
Orlando, Florida 32816-0177
USA
Telephone: 407-249-6100

(8) PROJECT 2851-SIF - Draft Military Standard Simulator Data Base (SSDB) Interchange Format (SIF) for High Detail Input/Output (SIF/HDI) and Distributed Processing (SIF/DP).

This draft can be obtained by writing to:

Wright Patterson Air Force Base
ARC/YP-C
Dayton, Ohio 45433

(9) SI - Le Syste'me International d'Unite's: See for example:
(A) "The Complete Metric System with the International System of Units, (SI)", A.L. LeMaraic & John P. Ciaramella, Abbey Books, 1973.

(B) "Guide for the Use of the International System of Units: The Modernized Metric System", A.O. McCoubrey, Sep. 91, 39p NIST/SP-811. Also available from Supt. of Docs. as SN003-003-03113-5.

(C) "Interpretation of the SI for the United States and Metric Conversion Policy for Federal Agencies", B.N. Taylor, Oct 91, 20p NIST/SP814. Also available from Supt. of Docs.

(10) SIMNET - BBN RPT. 7627 - The SIMNET Network and Protocols

This is available from:

National Technical Information Service (NTIS) 5285 Port Royal Road
Springfield, VA 22161
(703) 487-4650
Order: ADA244220

Note: The latest version of the General Reference documents should be used.

This page intentionally left blank.

3. Definitions

3.1 Acronyms Used in this Standard. The acronyms used in this standard are defined as follows:

(1) BBN - Bolt, Beranek and Newman

(2) CADIS - Communication Architecture for Distributed Interactive Simulation

(3) DARPA - Defense Advanced Research Projects Agency

(4) DIS - Distributed Interactive Simulation

(5) IEEE - Institute of Electrical and Electronic Engineers

(6) IST - Institute for Simulation and Training

(7) OSI - Open Systems Interconnection

(8) PDU - Protocol Data Unit

(9) SI - The official abbreviation for "Le Syste'me International d'Unite's": A universal system of metric weights and measures adopted in 1960 by the international authority on the metric system, the Comfe'rence G'en'erale de Polds et Measures (CGPM)

(10) SIMNET - Simulator Networking

(11) STRICOM - Simulation, Training and Instrumentation Command

(12) WGS 84 - World Geodetic System 1984

(13) UTC - Universal Coordinated Time

3.2 Terms

(1) **Application Layer (Layer 7).** The layer of the OSI reference model (ISO 7498) that provides the means for simulation applications to access and use the network's communications resources.

(2) **Articulated Part.** A visible part of a simulated entity that is able to move relative to the entity itself.

(3) **Attached Part.** A visible part of a simulated entity that may or may not be present. For example, a bomb on an aircraft wing station.

(4) **Ballistic Munition.** Any munition that follows a predetermined ballistic trajectory.

(5) **Best Effort Service.** A communication service in which transmitted data is not acknowledged. Such data typically arrives in order, complete and without errors. However, if an error occurs, nothing is done to correct it (e.g., there is no retransmission).

(6) **Bit.** The smallest unit of information in the binary system of notation.

(7) **Dead Reckoning.** A method for the estimation of the position/orientation of an entity based on a previously known position/orientation and estimates of time and motion.

(8) **Distributed Interactive Simulation (DIS).** A time and space coherent synthetic representation of world environments designed for linking the interactive, free play activities of people in operational exercises. The synthetic environment is created through real-time exchange of data units between distributed, computationally autonomous simulation applications in the form of simulations, simulators, and instrumented equipment interconnected through standard computer communicative services. The computational simulation entities may be present in one location or may be distributed geographically.

(9) **Emitter.** A device that is able to discharge detectable electromagnetic or acoustic energy.

(10) **Entity.** See Simulation Entity.

(11) **Entity Coordinate System.** Location with respect to a simulation entity is described by an entity coordinate system. See 1.2.2.

(12) **Euler Angles.** A set of three angles used to describe the orientation of an entity as a set of three successive rotations about three different orthogonal axes (x, y, and z). The order of rotation is first about z by angle Ψ (psi), then about the new y by angle Θ (theta), then about the newest x by angle Φ (phi) (see Figure 1-3). Angles Ψ and Φ range between $\pm\pi$, while angle Θ ranges only between $\pm\pi/2$ radians. These angles specify the successive rotations needed to transform from the world coordinate system to the entity coordinate system. The positive direction of rotation about an axis is defined by the right-hand rule.

(13) **Exercise.** See Simulation Exercise.

(14) **Fidelity.** A measure of the realism of a simulation.

(15) **Field.** A series of contiguous bits treated as an instance of a particular data type that may be part of a higher level data structure.

(16) **Guise.** A function that provides the capability for an entity to be viewed with one appearance by one group of participants, and with another appearance by another group.

(17) **Multicast.** A transmission mode in which a single message is sent to multiple network destinations, i.e., one-to-many.

(18) **Network Management.** The collection of administrative structures, policies, and procedures that collectively provide for the management of the organization and operation of the network as a whole.

(19) **Node.** A general term denoting either a switching element in a network or a host computer attached to a network.

(20) **Octet.** A sequence of eight bits, usually operated upon as a unit.

(21) **Protocol.** A set of rules and formats (semantic and syntactic) that determine the communication behavior of simulation applications.

(22) **Protocol Data Unit.** A unit of data that is passed on a network between simulation applications.

(23) **Real-Time.** An event or data transfer in which, unless accomplished within an allotted amount of time, the accomplishment of the action has either no value or diminishing value.

(24) **Real-World Time.** The standard Greenwich Mean Time. It is the actual time in Greenwich, England.

(25) **Reliable Service.** A communication service in which the number and type of errors that the user finds in the data are acceptable for the application. Reliable communication may require specific mechanisms in order to achieve the user's requirements: error detection and notification, such as bit errors based on a too-high bit error rate as defined by the user; or error detection and correction from PDU errors, such as bit errors, duplicated PDUs, missing PDUs, or out-of-sequence PDUs.

(26) **Right-Hand Rule.** Positive rotation is clockwise when viewed toward the positive direction along the axis of rotation.

(27) **Simulation Application.** The executing software on a host computer that generates one or more simulation entities. The simulation application represents or "simulates" real-world phenomena for the purpose of training or experimentation. Examples of simulation applications include manned vehicle simulators, computer generated forces, and computer interfaces between a DIS network and real equipment. The simulation application receives and processes information concerning entities created by peer simulation applications through the exchange of DIS PDUs. More than one simulation application may simultaneously execute on a host computer. The simulation application is the application layer protocol entity that implements the protocol defined in this document. This document will use the term "simulation application" to avoid confusion between protocol entities and simulation entities. The term "simulation" may also be used in place of simulation application.

(28) **Simulation Entity.** An element of the synthetic environment that is created and controlled by a simulation application through the exchange of DIS PDUs. Examples of types of simulated entities are: tank, submarine, carrier, fighter aircraft, missiles, bridges, or other elements of the synthetic

environment. It is possible that a simulation application may be controlling more than one simulation entity. Simulation entities may also be referred to as "entities" in this document.

(29) **Simulation Exercise.** An exercise that consists of one or more interacting simulation applications. Simulations participating in the same simulation exercise share a common identifying number called the Exercise Identifier. These simulations also utilize correlated representations of the synthetic environment in which they operate.

(30) **Simulation Management.** A mechanism that provides centralized control of the simulation exercise. Functions of simulation management include: start, restart, maintenance, shutdown of the exercise, and collection and distribution of certain types of data.

(31) **Simulation Time.** The time inside the simulation. It is the Greenwich Mean Time of the virtual world of the DIS exercise.

(32) **Tracked Munition.** A munition for which tracking data is required. A tracked munition's flight path is represented by Entity State PDUs.

(33) **Unicast.** A transmission mode in which a single message is sent to a single network destination, i.e., one-to-one.

(34) **World Coordinate System.** The right-handed geocentric Cartesian system. The shape of the world is described by the WGS 84 standard. The origin of the world coordinate system is the centroid of the earth. The axes of this system are labeled X, Y, and Z, with the positive X-axis passing through the Prime Meridian at the Equator, with the positive Y-axis passing through 90 degrees East longitude at the Equator and the positive Z-axis passing through the North pole.

4. General Requirements

4.1 Introduction. This section contains requirements concerning the content and use of PDUs in DIS exercises.

4.1.1 DIS Exercise. A DIS exercise shall consist of one or more interacting simulation applications. The DIS PDUs issued by all simulation applications participating in the same exercise shall share one identifying number called the Exercise Identifier.

4.1.2 Issuing Entity. In a DIS exercise, simulation applications may represent one or more entities. The simulation application shall issue PDUs for each of the entities that it simulates. PDUs shall never be issued in advance of the occurrence of the event or state communicated by the PDU. For reference, it is easier to state that the entity is issuing the PDU, rather than stating that the simulation application that is simulating the entity is issuing the PDU. Therefore, the phrase "entity issues" shall mean "the simulation application representing the entity issues."

4.2 Issuance of PDUs. PDUs shall be issued according to the requirements specified in the paragraphs that follow.

4.3 Receipt of PDUs. Upon receipt of PDUs, simulation applications shall act as specified in the paragraphs that follow. Unless otherwise stated, the actions described in these paragraphs shall apply to all the simulation entities to which the PDU is addressed. Depending on the PDU type, the PDU can be addressed to a particular entity or to all entities within an exercise. Simulation applications shall accommodate out-of-order delivery of PDUs.

4.4 Protocol Data Units for DIS. The following paragraphs shall establish the content and the procedure for use of PDUs in a DIS exercise.

4.4.1 Protocol Data Unit Header. A PDU header shall be the first part of each PDU. The header shall specify the identification number associated with the DIS exercise, the protocol version, the type of protocol data unit that follows, the time stamp, and the length of the PDU (see also 5.3.15).

4.4.1.1 Protocol Version. The PDU header shall specify the version of DIS protocol to which the PDU pertains.

4.4.1.2 DIS Exercise Identification. Each DIS exercise shall be distinguished from other exercises by the use of an Exercise Identifier. An identifier that is currently not in use on the network shall be assigned.

4.4.1.3 Types of PDUs. This standard defines the following 27 PDUs:

(1) Entity State PDU
(2) Fire PDU
(3) Detonation PDU
(4) Service Request PDU
(5) Resupply Offer PDU
(6) Resupply Received PDU
(7) Resupply Cancel PDU
(8) Repair Complete PDU
(9) Repair Response PDU
(10) Collision PDU
(11) Create Entity PDU
(12) Remove Entity PDU
(13) Start/Resume PDU
(14) Stop/Freeze PDU
(15) Acknowledge PDU
(16) Action Request PDU
(17) Action Response PDU
(18) Data Query PDU
(19) Set Data PDU
(20) Data PDU
(21) Event Report PDU
(22) Message PDU
(23) Emission PDU
(24) Laser PDU
(25) Transmitter PDU
(26) Signal PDU
(27) Receiver PDU

In addition to these PDUs, other PDUs may be added to this standard in the future. PDU types have been reserved for PDU growth. PDU types in the range of 129 through 255 have been reserved for experimental purposes.

4.4.1.4 Time Stamp. This field shall specify the time at which the data in the PDU is valid.

4.4.1.5 Length of PDU. The Length field shall specify the number of octets between and including the first and last octet of the PDU.

4.4.2 Entity Information. Information associated with the appearance and location of an entity shall be communicated in a DIS exercise through the use of an Entity State PDU (see also 5.4.3.1).

4.4.2.1 Entity State PDU. The Entity State PDU shall communicate information about an entity's state. This includes state information that is necessary for the receiving simulation applications to represent the issuing entity in the simulation applications' own simulation.

4.4.2.1.1 Information Contained in the Entity State PDU. The Entity State PDU shall contain the following information:

(1) The identification of the entity that issued the PDU
(2) The identification of the force to which the entity belongs
(3) The issuing entity's specific entity type
(4) The issuing entity's alternate entity type for use with the Guise function (see 3.2-16)
(5) Information about the location of the entity in the simulated world and its orientation. This information includes:
  (a) The location with respect to the world
  (b) The velocity (the rate at which its location is changing)
  (c) Orientation
  (d) The dead reckoning parameters that should be employed when extrapolating the position of this entity. Values in this field shall include dead reckoning algorithm in use, linear acceleration, and angular velocity. Other values for this field are currently undefined.
(6) The information required for representation of the entity's visual appearance. This information includes:
  (a) The appearance of the entity (for example, normal, smoking, on fire, producing a dust cloud, etc.)
  (b) Markings
  (c) The number of articulation parameters and the parameter values to represent orientation of articulated parts
  (d) The presence of attached parts or stores
(7) The capabilities of the entity. Defined capabilities include:
  (a) Resupply
  (b) Repair

4.4.2.1.2 Dead Reckoning. A method of position/orientation estimation called dead reckoning shall be employed to limit the rate at which Entity State PDUs are issued. For more information on dead reckoning, see Section 7 in Document IST-CR-93-02.

4.4.2.1.2.1 Dead Reckoning and the Issuing Entity. Each simulation application shall maintain two models of each entity it is representing. One model shall be a high fidelity model of the entity. The other model shall be a dead reckoning model of the entity it is representing. Certain thresholds shall be established as criteria for determining if the entity's actual position/orientation has varied an allowable amount from its dead reckoned position/orientation. When the entity's actual position/orientation has varied from the dead reckoned position/orientation by more than a threshold value, the simulation application shall issue an Entity State PDU to communicate the entity's actual position to other simulation applications. The simulation application shall also use the same information communicated to other simulation applications to update its dead reckoning model of its entity.

4.4.2.1.2.2 Dead Reckoning and the Receiving Entity. Each simulation application shall maintain a dead reckoning model of the position/orientation of entities that are of interest (within sight or range) as specified by the dead reckoning model in use. The dead reckoned position/orientation of other entities shall be used to display their position/ orientation in a simulator's visual or sensor displays. When the simulation application receives a new update from one of the entities it is dead reckoning, it shall correct its dead reckoning model and base its estimations on the most recent position/orientation, velocity, and acceleration information. Smoothing techniques may be used to eliminate jumps that occur in a visual display when the dead reckoning position/orientation of an entity is corrected using more recent position/orientation data.

4.4.2.1.2.3 Dead Reckoning Algorithms. Required dead reckoning algorithms for use with this standard are included in Section 7 in Document IST-CR-93-02. To allow for parameters associated with the dead reckoning algorithm in use, a field has been set aside in the Entity State PDU for dead reckoning parameters.

4.4.2.1.3 Issuance of the Entity State PDU. The Entity State PDU shall be issued by an entity when one or more of the following conditions exist:

(1) The discrepancy between an entity's actual state (as determined by its own high fidelity model) and its dead reckoned state (state using specified dead reckoning algorithms) exceeds a predetermined threshold (see Section 7 in Document IST-CR-93-02 concerning dead reckoning and threshold values). This threshold includes changes in position/orientation information and articulation parameter information.

(2) A change in the entity's appearance occurs. This change may include beginning to burn or smoke.

(3) A predetermined length of time has elapsed since the issuing of the last Entity State PDU. This value may be established at exercise start or during the exercise. The mechanism by which this value is established is outside the scope of the protocol. If no value is established, the default value shall be five seconds. A default tolerance of $\pm 10\%$ shall apply to this value. The tolerance value may be set by a mechanism outside the scope of this protocol.

The Entity State PDU shall be issued using a real-time, best effort, multicast communication service. The PDU shall be sent to all simulation applications participating in the same exercise.

4.4.2.1.4 Receipt of the Entity State PDU. Upon receipt of the Entity State PDU, a simulation application shall determine that the PDU contains newer information than that currently being used to model the transmitting entity. If so, the simulation application shall use the information contained therein to model the position, orientation, and appearance of the entity that issued the PDU. Otherwise the PDU shall be discarded. If a predetermined length of time has elapsed since the last Entity State PDU representing that a given entity was received, the simulation application shall remove the model of that entity from its simulation. The length of time shall be established at exercise start, or may be changed during the exercise. If no value is established, the default value for this time shall be the value of the Entity State PDU issuance interval (see 4.4.2.1.3, condition 3) multiplied by 2.4

4.4.2.1.5 Guise. The purpose of the Guise function is to allow DIS participants to serve in the Opposing Force role while operating friendly simulations. This function allows both sides of an engagement to see their team members as friendly and the opposing forces as hostile.

The Entity Type field shall be used by members of the issuing entity's team to display the issuing entity. The Alternate Entity Type field shall be used by members of all other teams to display the issuing entity. The Force ID determines the team membership of the issuing entity.

4.4.2.1.5.1 Issuing Entity Actions. If the Guise function is used, an issuing entity shall perform the following actions:

(1) The entity shall assign the entity type code for its friendly force appearance to the Entity Type field of the Entity State PDUs it issues.
(2) The entity shall assign the entity type code for its opposing force appearance to the Alternate Entity Type field of the Entity State PDUs it issues.
(3) In the case of neutral entities, the same type code shall be assigned to both the Entity Type field and the Alternate Entity Type field.

If the Guise function is not used, the same entity type shall be assigned to both the Entity Type field and the Alternate Entity Type field.

4.4.2.1.5.2 Receiving Entity Actions. The receiving entity shall interpret Entity Type and Alternative Entity Type fields. Upon receiving an Entity State PDU from an entity that is of interest, the receiving entity shall check the value of the Force ID of the entity that issued the PDU. The following conditions apply:

(1) If the Force ID is the same as Receiving Entity, the issuing entity should be displayed by its friendly force appearance found in the Entity Type field.
(2) If the Force ID is different from the Receiving Entity, the issuing entity should be displayed by its opposing force appearance found in the Alternate Entity Type field.

4.4.2.1.5.3 Example of Guise Function. The Guise function might be employed in the following manner: In a simulation exercise, the friendly force appearance is an M1 tank, the opposing force appearance is a T-72. All participants would use an entity type of M1 and an alternate entity type of T-72 in the Entity State PDUs they issue.

During the exercise, members of Team 1 receiving Entity State PDUs from their own team will depict them using the friendly force appearance M1. Entity State PDUs received from members of Team 2 will be depicted by members of Team 1 as T-72s by using the opposing force appearance.

Likewise, members of Team 2 receiving Entity State PDUs from their own team will depict them using the friendly force appearance M1. Entity State PDUs received from members of Team 1 will be depicted by members of Team 2 as T-72s.

4.4.3 Weapons Fire. Information associated with the representation of weapons effects in a DIS exercise shall be communicated through the use of two PDUs: the Fire PDU and the Detonation PDU (see also 5.4.4.1).

4.4.3.1 Representation of Weapons Fire in DIS. Representation of weapons fire in a DIS exercise shall consist of the following sequence of events:

(1) An entity fires a weapon. The firing of a weapon shall be communicated through the use of a Fire PDU.
(2) The munition launched shall be modeled by the firing entity's simulation application. If the munition is a munition for which tracking data is required, it shall be assigned a unique entity identifier by the firing entity's simulation application. In addition to issuing the Fire PDU, the simulation application modeling the munition's behavior shall issue Entity State PDUs for the munition according to the procedures for the use of the Entity State PDU (see 4.4.2). The munition, therefore, is represented as an entity. If tracking data is not required for the munition, the munition entity identifier shall be zero. The decision as to whether or not a munition is trackable is outside the scope of this standard.

(3) The impact or detonation of a munition represents the end of its path and the end of the existence of a munition entity. This event shall be represented by a Detonation PDU. A Detonation PDU shall also be used to terminate undetonated munitions (see 4.4.3.3.6) that do not impact or detonate.

4.4.3.2 Fire PDU. The Fire PDU shall be used to communicate information associated with the firing of a weapon.

4.4.3.2.1 Information Contained in the Fire PDU. The Fire PDU shall contain the following information:

(1) The identification of the entity issuing the PDU
(2) The identification of the intended target entity if known to the simulation application, zero otherwise
(3) The identification of tracked munitions (munitions not tracked have an identification value of zero)
(4) The identification of the specific event marked by the firing of an entity's weapon
(5) The information required for representation of the path and impact of the munition. This information includes:
  (a) The location from which the munition was launched or fired
  (b) The type of munition fired
  (c) The warhead of the munition (if applicable, zero otherwise)
  (d) The fuze employed by the munition (if applicable, zero otherwise)
  (e) The quantity and rate at which it was fired
  (f) The initial velocity of munition fired
  (g) The range the firing entity's fire control system has assumed for computing the fire control solution

4.4.3.2.2 Issuance of the Fire PDU. The Fire PDU shall be issued by an entity at the moment it fires a weapon.

The Fire PDU shall be issued using a real-time, best effort, multicast communication service. The PDU shall be sent to all simulation applications participating in the same exercise.

4.4.3.2.3 Single Rounds and Bursts of Fire. If the firing of the weapon represents a single round, the quantity field of the Fire PDU shall contain the value one, and the rate field shall contain the value zero. If the firing of a weapon or a group of weapons oriented in the same direction represents multiple rounds, the fields shall contain the quantity of munition fired and the rate at which it was fired, respectively.

4.4.3.2.4 Receipt of the Fire PDU. Upon receipt of a Fire PDU, a simulation application shall use the information therein to represent any necessary visual and aural effects produced by the firing of the weapon, whether it be a muzzle flash, noise, or smoke.

4.4.3.3 Detonation PDU. The Detonation PDU shall be used to communicate information associated with the impact or detonation of a munition.

4.4.3.3.1 Information Contained in the Detonation PDU. The Detonation PDU shall contain the following information:

(1) The identification of the entity issuing the PDU
(2) The identification of the target entity if it impacted an entity, zero otherwise
(3) The identification of tracked munitions (munitions not tracked shall have an identification value of zero)
(4) The identification of the fire event responsible for the detonation. This number shall be the same as the event identification number assigned to the corresponding Fire PDU. If the detonation is not preceded by a corresponding fire event, then the event identifier field of the event identifier record shall be zero (e.g., land mines detonation).
(6) The information required for representation of the impact or detonation of the munition. This information includes:
    (a) The location with respect to the world
    (b) The type of munition fired
    (c) The warhead of the munition (if applicable, zero otherwise)
    (d) The fuze employed by the munition (if applicable, zero otherwise)
    (e) The quantity and rate at which it was fired
    (f) The velocity just before detonation/impact
    (g) The location of detonation with respect to the target entity
    (h) The detonation result
    (i) Articulation parameters for any articulated parts of the target entity affected by the detonation

4.4.3.3.2 Issuance of the Detonation PDU. The Detonation PDU shall be issued by a simulation application at the moment that a munition being modeled by that simulation application impacts or detonates. If the munition neither impacts nor detonates, the controlling simulation application shall issue a Detonation PDU with a detonation result of "none" when the controlling simulation application has ceased to model the munition.

The Detonation PDU shall be issued using a real-time, best effort, multicast communication service. The PDU shall be sent to all simulation applications participating in the same exercise.

4.4.3.3.3 Interpretation of Detonation Result and Inclusion of Entity Identifier. If the impact or detonation is known to have affected only a specific entity, a result of "entity impact" is used if the entity is directly contacted by the munition. Otherwise, a detonation result of "entity proximate detonation" is used. In either case, the firing entity shall communicate the entity identifier (Target Entity ID) in the Detonation PDU. The location of the impact or detonation in the entity coordinates of the affected entity shall also be included.

If the impact or detonation is known to have affected only the terrain, a result of "ground impact" is used if the terrain is directly contacted by the munition. Otherwise a detonation result of "ground proximate detonation" is used. In these cases the location of the impact or detonation shall be communicated in world coordinates.

If neither a specific entity nor the terrain is all that is affected by the munition, a detonation result of "detonation" is used if the munition detonates, and a detonation result of "none" is used if it does not. In either case, the terminal location of the munition shall be communicated in world coordinates.

4.4.3.3.4 Inclusion of Articulated Part Parameters. When the firing entity determines that its round has impacted an articulated part on the target entity, the firing entity shall include the articulation parameters of the affected articulated part (at the time of impact) in the Detonation PDU. If the articulated part is not directly connected to the base model, all intervening articulations shall be included in the Detonation PDU. The articulation parameters shall be represented in the same form as in the Entity State PDU.

4.4.3.3.5 Termination of the Existence of Munition Entities. The Detonation PDU shall indicate the termination of the existence of a munition entity. Upon receipt of the Detonation PDU, simulation applications shall stop modeling the munition.

4.4.3.3.6 Termination of Undetonated Munitions. If the munition neither impacts nor detonates, the controlling simulation application shall issue a Detonation PDU to indicate when it has ceased to model the munition. For example, if the munition has exceeded a certain range, the controlling simulation application shall issue a Detonation PDU and shall cease to model that munition. In the case of an undetonated munition, the location field of the Detonation PDU shall contain the location of the munition entity when the controlling simulation application stopped modeling the munition. The detonation result shall contain the value "none".

4.4.3.3.7 Receipt of the Detonation PDU. Upon the receipt of a Detonation PDU, a simulation application shall use the information therein to represent the visual and aural effects that may be produced by the detonation or impact of the munition. The receiving simulation application shall also use the information to determine damage that may have been received as a result of the detonation.

4.4.4 Logistics Support. Information associated with the representation of logistics support in a DIS exercise shall be communicated through the use of several PDUs: Service Request PDU, Resupply Offer PDU, Resupply Received PDU, Resupply Cancel PDU, Repair Complete PDU, and Repair Response PDU (see 5.4.4.2).

The procedures associated with logistics support require the definition of several timers. The value of these timers shall be established at exercise start and may be changed during the exercise. The mechanism by which these values are set and changed is outside the scope of this protocol. If no values are established, the default values for these timers shall be as follows:

(1) Resupply Receiver Timer 1: 5 seconds
(2) Resupply Receiver Timer 2: 1 minute
(3) Resupply Supplier Timer 1: 1 minute
(4) Repair Receiver Timer 1: 5 seconds
(5) Repair Supplier Timer 1: 12 seconds
(6) Repair Supplier Timer 2: 12 seconds

4.4.4.1 Procedure for Logistics Support. Logistics support in DIS shall be accomplished through a series of request and response messages between two entities. Two types of service have been defined for DIS: resupply and repair.

4.4.4.2 State Information for Resupply. The following paragraphs describe the different states and transitions for resupply service. An example of the resupply function is given in 4.4.4.8.

4.4.4.2.1 Receiving Entity. The receiving entity may be in one of three states:

(1) Ready State	A receiving entity is in the Ready state when it is not in either the Requesting state or the Receiving state.
(2) Requesting State	A receiving entity is in the Requesting state when it has requested supplies and has not received a reply to its request.
(3) Receiving State	A receiving entity is in the Receiving state when it has been offered supplies and is in the process of receiving them.

The behavior of the receiving entity during resupply service is shown in Fig 4-1.

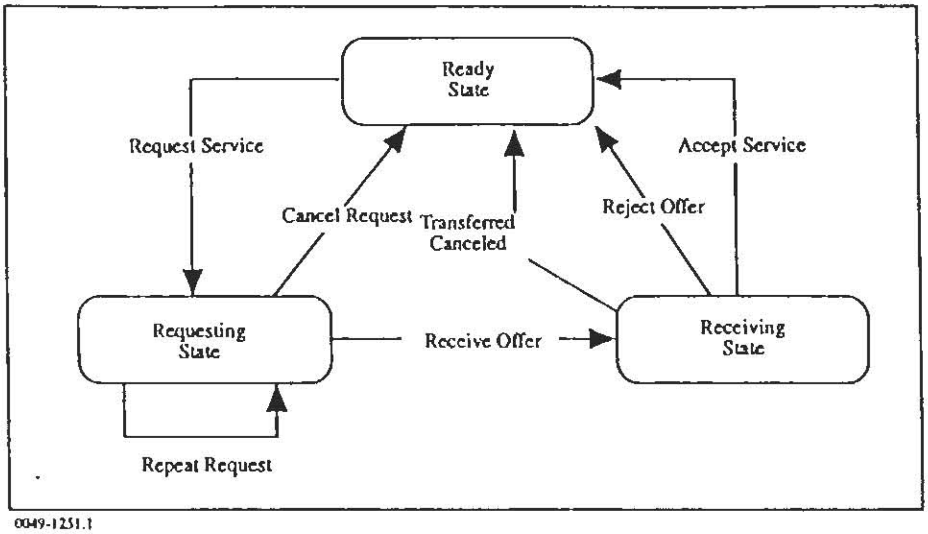


Fig 4-1
Receiving Entity Behavior During Resupply

Transition	Condition and Actions
Request Service	When conditions for resupply are met, the entity shall issue a Service Request PDU (see 4.4.4.4). The entity shall proceed from the Ready state to the Requesting state and Resupply Receiver Timer 1 shall be set.[1]
Cancel Request	When conditions for resupply are no longer met or when a Resupply Cancel PDU (4.4.4.7) is received from a supplying entity, Resupply Receiver Timer 1 shall be canceled and the entity shall proceed from the Requesting state to the Ready state.

The fact that an entity is able to provide resupply or repair service is indicated in the Capabilities field in its ES PDUs.

Repeat Request	When Resupply Receiver Timer 1 expires, the Service Request PDU shall be reissued and the timer shall be reset.
Receive Offer	When a Resupply Offer PDU (see 4.4.4.5) is received, Resupply Receiver Timer 1 shall be canceled and Resupply Receiver Timer 2 shall be set to the period of time required for receiving some portion of the offered supplies. The entity shall proceed from the Requesting state to the Receiving state.
Reject Offer	When conditions for resupply are no longer met, a Resupply Cancel PDU shall be issued, Resupply Receiver Timer 2 shall be canceled, and the entity shall proceed from the Receiving state to the Ready state.
Accept Service	When Resupply Receiver Timer 2 expires, the count of supplies on board shall be incremented and a Resupply Received PDU (see 4.4.4.6) shall be issued. The entity shall then proceed from the Receiving state to the Ready state.
Transfer Canceled	When a Resupply Cancel PDU (see 4.4.4.7) is received, the ongoing transfer shall be canceled, Resupply Receiver Timer 2 shall be canceled, and the entity shall proceed from the Receiving state to the Ready state. No supplies (from the canceled transfer) shall be transferred.

4.4.4.2.2 Supplying Entity. The supplying entity may be in one of two states:

(1) Ready State	A supplying entity is in the Ready state when it is able to receive a request for supplies and is able to offer supplies to a receiving entity.
(2) Offering State	A supplying entity in the Offering state has made an offer of supplies and is waiting for the receiving entity to indicate the quantity of supplies it has taken.

The behavior of the supplying entity during resupply service is shown in Fig 4-2.

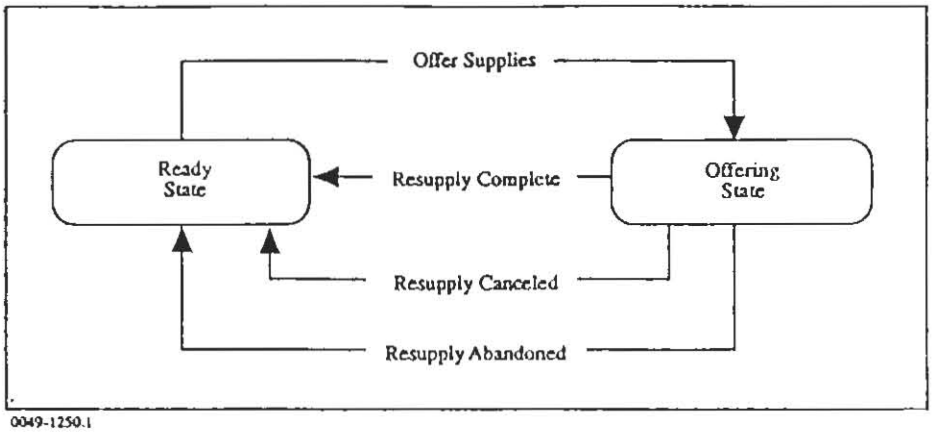


Fig 4-2
Supplying Entity Behavior During Resupply

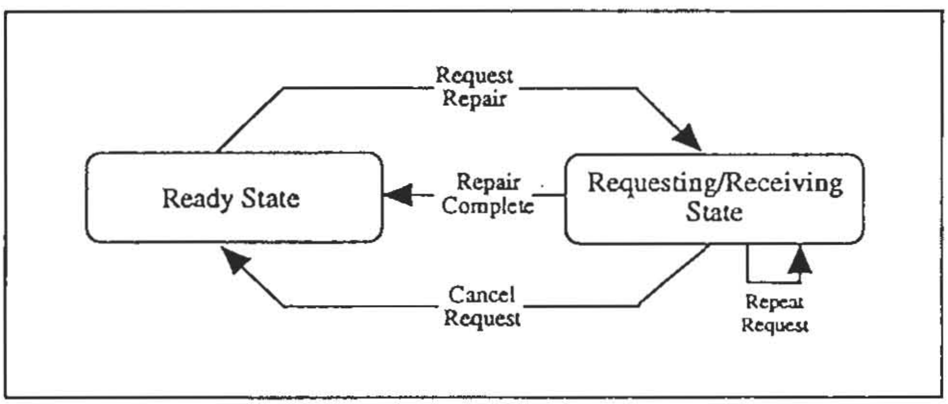
Transition	Condition and Actions
Offer Supplies	When a Service Request PDU (see 4.4.4.4) is received, a Resupply Offer PDU (see 4.4.4.5) shall be issued, Resupply Supplier Timer 1 shall be set, and the entity shall proceed from the Ready state to the Offering state.
Resupply Complete	When a Resupply Received PDU (see 4.4.4.6) is received, Resupply Supplier Timer 1 shall be canceled. The count of supplies on board shall be decremented and the entity shall proceed from the Offering state to the Ready state.
Resupply Canceled	When a Resupply Cancel PDU (see 4.4.4.7) is received, Resupply Supplier Timer 1 shall be canceled. The count of supplies on board shall not change and the entity shall proceed from the Offering state to the Ready state.
Resupply Abandoned	When Resupply Supplier Timer 1 expires, the transfer shall be abandoned, the count of supplies on board shall not change, and the entity shall proceed from the Offering state to the Ready state.

4.4.4.3 State Information for Repair. The following paragraphs describe the different states and transitions for repair. An example of repair service is given in 4.4.4.12.

4.4.4.3.1 Receiving Entity. The receiving entity may be in one of two states:

(1) Ready State	A receiving entity is in the Ready state when it is able to request repairs from an entity with repair capabilities.
(2) Requesting/ Receiving State	A receiving entity is in the Requesting/Receiving Receiving state when it has requested repairs and has not received a response to its request.

The behavior of the receiving entity during repair service is shown in Fig 4-3.



0049-1440

Fig 4-3
Receiving Entity Behavior During Repair

Transition	Condition and Actions
Request Repair	When conditions for repair service are met, the entity shall issue a Service Request PDU (see 4.4.4.4). Repair Receiver Timer 1 shall be set. The entity shall proceed from the Ready state to the Requesting/Receiving state.
Repair Complete	When a Repair Complete PDU (see 4.4.4.9) is received, the entity shall issue a Repair Response PDU (see 4.4.4.10), Repair Receiver Timer 1 shall be canceled, and the entity shall proceed from the Requesting/Receiving state to the Ready state.

Cancel Request — When conditions for repair service are no longer met, the receiving entity shall cease to issue Service Request PDUs, Repair Receiver Timer 1 shall be canceled, and the entity shall proceed from the Requesting/Receiving state to the Ready state.

Repeat Request — When Repair Receiver Timer 1 expires, the Service Request PDU shall be reissued and the timer reset. The entity shall remain in the Requesting/Receiving state.

4.4.4.3.2 Repairing Entity. The repairing entity may be in one of two states:

(1) Ready State — A repairing entity is in the Ready state when it is able to offer repairs to a receiving entity.

(2) Offering State — A repairing entity is in the Offering state when it has received a request for repairs and is responding to the request.

(3) Repair Completed State — A repairing entity is in the Repair Completed State when it has completed a repair and it is waiting for a response.

The behavior of the repairing entity during repair service is shown in Fig 4-4.

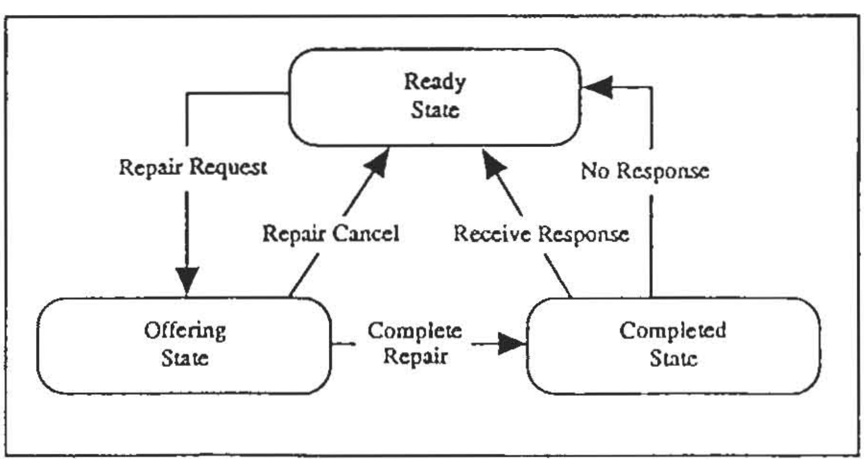


Fig 4-4
Repairing Entity Behavior During Repair

Transition	Conditions and Actions
Repair Request	When the repairing entity has received a Service Request PDU (see 4.4.4.4) and has identified itself as the intended repairing entity, Repair Supplier Timer 1 shall be set. Every time a Service Request PDU is received from the entity requesting the repair service, Repair Supplier Timer 1 shall be reset. The repairing entity shall then proceed from the Ready state to the Offering state.
Complete Repair	When the repair is complete, the entity shall issue a Repair Complete PDU (see 4.4.4.9), shall cancel Repair Supplier Timer 1, shall set Repair Supplier Timer 2, and shall proceed from the Offering state to the Repair Completed state.
Receive Response	When a Repair Response PDU is received from the receiving entity, entity shall cancel the Repair Supplier Timer 2 and shall proceed from the Repair Completed State to the Ready state.
No Response	When Repair Supplier Timer 2 has expired, the entity shall proceed from the Repair Completed state to the Ready state.
Repair Canceled	When Repair Supplier Timer 1 has expired, the entity shall proceed from the Offering state to the Ready state.

4.4.4.4 Service Request PDU. The Service Request PDU shall be used to communicate information associated with one entity requesting a service from another.

4.4.4.4.1 Information Contained in the Service Request PDU. The Service Request PDU shall contain the following information:

(1) The identification of the entity issuing the PDU
(2) The identification of the entity that is able to provide the service required by the requesting entity
(3) The type of service being requested. Services defined are:
  (a) Resupply
  (b) Repair
(4) The number and types of supplies if the service required is resupply

4.4.4.4.2 Issuance of the Service Request PDU. The Service Request PDU shall be issued by an entity requesting logistics support when appropriate conditions[1] exist.

The Service Request PDU shall be issued using a best effort, multicast communication service.

4.4.4.4.3 Receipt of the Service Request PDU. Upon receipt of a Service Request PDU, the entity that receives the PDU shall respond in one of the following ways:

(1) If the service requested is resupply and the entity that receives the PDU is able to provide the needed supplies, the supplying entity shall issue a Resupply Offer PDU (see 4.4.4.5).
(2) If the service requested is repair, the repairing entity shall simulate the needed repairs in the following manner: The repair process is allowed to proceed as long as the repairing entity continues to receive Service Request PDUs. If Service Request PDUs cease to be received and are not seen for the period of Repair Supplier Timer 1, the repairing entity shall assume that conditions for repair no longer exist and therefore shall abort the process.
(3) If the service requested is resupply and the entity receiving the PDU is unable to provide the supplies requested, then the supplying entity shall issue a Resupply Cancel PDU (see 4.4.4.7).

4.4.4.5 Resupply Offer PDU. A Resupply Offer PDU shall be used to communicate the offer of supplies by a supplying entity to a receiving entity.

4.4.4.5.1 Information Contained in the Resupply Offer PDU. The Resupply Offer PDU shall contain the following information:

(1) The identification of the entity requesting resupply
(2) The identification of the supplying entity which issued the PDU
(3) The number of types of supplies that the supplying entity is able to provide
(4) The supply types available and the amount of each

4.4.4.5.2 Issuance of the Resupply Offer PDU. The Resupply Offer PDU shall be issued by an identified supplying entity that

---

[1] Appropriate conditions include internal conditions such as certain crew actions in the simulator, as well as external conditions such as conditions existing in the simulated would entities not destroyed, or being within a certain distance.

has received a Service Request PDU (see 4.4.4.4) requesting resupply service.

The Resupply Offer PDU shall be issued using a best effort, multicast communication service.

4.4.4.5.3 Receipt of the Resupply Offer PDU. Upon receipt of a Resupply Offer PDU, the receiving entity shall proceed from the Requesting state to the Receiving state. When receipt of the supplies is complete, the receiving entity shall respond by issuing a Resupply Received PDU.

4.4.4.6 Resupply Received PDU. A Resupply Received PDU shall be used to acknowledge the receipt of supplies by the receiving entity.

4.4.4.6.1 Information Contained in the Resupply Received PDU. The Resupply Received PDU shall contain the following information:

(1) The identification of the entity requesting resupply
(2) The identification of the supplying entity
(3) The number of types of supplies that the supplying entity is able to provide
(4) The supply types available and the amount of each taken by the receiving entity

4.4.4.6.2 Issuance of the Resupply Received PDU. The Resupply Received PDU shall be issued by an identified receiving entity to indicate the supplies actually transferred from the supplying entity to the receiving entity.

The Resupply Received PDU shall be issued using a best effort, multicast communication service.

4.4.4.6.3 Receipt of the Resupply Received PDU. Upon receipt of a Resupply Received PDU, the supplying entity shall decrement the number of supplies on board and shall proceed to the Ready state.

4.4.4.7 Resupply Cancel PDU. The Resupply Cancel PDU shall be used to communicate the canceling of a resupply service provided through logistics support.

4.4.4.7.1 Information Contained in the Resupply Cancel PDU. The Resupply Cancel PDU shall contain the following information:

(1) The identification of the entity receiving supplies
(2) The identification of the entity providing supplies

4.4.4.7.2 Issuance of the Resupply Cancel PDU. The Resupply Cancel PDU shall be issued by either the receiving entity or supplying entity at any time during resupply to cancel the resupply service.

The Resupply Cancel PDU shall be issued using a best effort, multicast communication service.

4.4.4.7.3 Receipt of the Resupply Cancel PDU. The entity receiving the Resupply Cancel PDU shall cancel its timers and return to the Ready state. No supplies (for the canceled transaction) are transferred.

4.4.4.8 Example of Resupply Service. If the service requested by the receiving entity is resupply, a scenario such as the following may take place:

A resupply interaction begins when a simulation application determines that one of its entities is in a state such that it should be resupplied. For example, a ground vehicle might come to a stop and open its refueling port. When this occurs, the simulation application issues a Service Request PDU to a potential resupplying entity and sets Resupply Receiver Timer 1. Potential resuppliers may be identified by their proximity, state, and the content of the capabilities field (as determined from their Entity State PDUs). This event is shown as the "Request Service" transition in Figure 4-1.

A simulation entity that receives a Service Request PDU and is identified as the specified supplying entity responds by offering some portion of whatever supplies are currently loaded on board. This condition is shown in Fig 4-2 (see 4.4.4.2.2) as a transition from the Ready state to the Offering state. Meanwhile, the receiving entity reissues its Service Request PDU every time Resupply Receiver Timer 1 expires until such an offer is forthcoming. The offer takes the form of a Resupply Offer PDU issued by the supplying entity. The supplies offered should be a subset of those possessed by the supplying entity, and a subset of those requested by the receiving entity.

Upon receiving the offer of supplies, the receiving entity changes from the Requesting state to the Receiving state. The receiving entity then has until the Resupply Receiver Timer 2 expires to acknowledge the receipt of those supplies by returning to the supplying entity a Resupply Received PDU listing the exact supplies taken. The receiving entity need not accept all of the supplies offered, but instead can indicate in its receipt how many it did accept. After delaying up to the time which Resupply Receiver Timer 2 expires, the receiving entity issues its Resupply

Received PDU and returns to the Ready state. When the supplying entity receives the Resupply Received PDU, it also returns to the Ready state, and the procedure is complete.

The time required to return the Resupply Received PDU, and the quantity of supplies reported by that PDU as taken determine the rate at which the supplying entity and the receiving entity are able to transfer munitions. For example, an M1 tank obtaining 105 mm shells from an ammunition supply truck might acknowledge receipt of a single round after 40 seconds; this results in a simulated rate of resupply for the M1 tank of one round every 40 seconds.

Throughout the transfer process, both the receiving entity and the supplying entity continue to monitor the conditions necessary for the transfer. If any of these conditions cease to hold, either entity can abort the transfer by issuing a Resupply Cancel PDU, with the result that no supplies are transferred (for the transfer that was in process). Alternatively, the receiving entity can terminate the transfer early but accept some of the supplies offered by issuing a Resupply Received PDU for the partial load. Finally, if the supplying entity waits in the Offering state for the entire period of Resupply Supplier Timer 1 but receives no Resupply Received PDU (perhaps the receiving entity has withdrawn from the exercise), it should return to the Ready state and assume that no supplies were taken.

4.4.4.9 Repair Complete PDU. The Repair Complete PDU shall be used by the repairing entity to communicate the repair that has been performed for the entity that requested repair service.

4.4.4.9.1 Information Contained in the Repair Complete PDU. The Repair Complete PDU shall contain the following information:

(1) The identification of the entity requesting repair service
(2) The identification of the entity providing the repair
(3) The repair performed by the repairing entity. Possible repairs are included in Section 2 in Document IST-CR-93-02.

4.4.4.9.2 Issuance of the Repair Complete PDU. The Repair Complete PDU shall be issued by a repairing simulation entity upon completion of a repair service requested by the receiving entity in a Service Request PDU (see 4.4.4.4).

The Repair Complete PDU shall be issued by using a best effort, multicast communication service.

4.4.4.9.3 Receipt of the Repair Complete PDU. Upon receipt of the Repair Complete PDU, the receiving entity shall issue a Repair Response PDU (see 4.4.4.10), and proceed from the Requesting/Receiving state to the Ready state.

4.4.4.10 Repair Response PDU. A Repair Response PDU shall be used by the receiving entity to acknowledge the receipt of a Repair Complete PDU (see 4.4.4.9).

4.4.4.10.1 Information Contained in the Repair Response PDU. The Repair Response PDU shall contain the following information:

(1) The identification of the entity requesting repair service
(2) The identification of the entity providing the repair
(3) The result of the repair

4.4.4.10.2 Issuance of the Repair Response PDU. The Repair Response PDU shall be issued by the entity receiving repair service upon receipt of a Repair Complete PDU from the repairing entity.

The Repair Response PDU shall be issued by using a best effort, multicast communication service.

4.4.4.10.3 Receipt of the Repair Response PDU. Upon receipt of the Repair Response PDU, the repairing entity shall note that the receiving entity has received the repair.

4.4.4.11 Cancellation of Repair Service. If the receiving entity intends to cancel the repair service before the repairs are completed, it shall cease to issue Service Request PDUs and shall return to the Ready state. The supplying entity that does not receive Service Request PDUs for the period of Repair Supplier Timer 1 shall abandon the repair service and shall return to the Ready state. If the supplying entity intends to cancel the repair service, it shall issue a Repair Complete PDU and shall list the repair result as "no repairs performed." The receiving entity shall respond with the issue of a Repair Response PDU indicating the repair result as "service canceled by the supplying entity."

4.4.4.12 Example of Repair Service. If the service requested by the receiving entity is repair, a scenario such as the following may take place:

A simulation application issues a Service Request PDU and sets Repair Receiver Timer 1 when it determines that one of its simulation entities is in need of repairs. The Service Request PDU is issued to an entity capable of performing repairs as determined by its proximity, state, and the content of the field capabilities conveyed in its Entity State PDU. This event is shown as the Request Repair transition in Figure 4-3.

If the repair process successfully runs to completion, the repairing entity may then accomplish the repair by issuing a Repair Complete PDU to notify the receiving entity of the repair, and returning to the Ready state. The receiving entity's simulator acknowledges the receipt of the Repair Complete PDU by returning a

Repair Response PDU. (This acknowledgment simply indicates that the repair was performed, not that the repair was appropriate or that the disabled vehicle has been made operational because of the repair.)

4.4.5 Collisions. Information associated with collisions between entities shall be communicated in a DIS exercise through the use of the Collision PDU (see also 5.4.4.3).

4.4.5.1 Collision PDU. The Collision PDU shall be used to communicate information about a collision between two simulated entities or between a simulated entity and another object in the simulated world (such as a cultural feature).

4.4.5.1.1 Information Contained in the Collision PDU. The Collision PDU shall contain the following information:

(1) The identification of the entity that issued the PDU
(2) The identification of the entity with which the issuing entity collided. If this ID number is unknown, the ID field shall contain zeros
(3) The identification of the specific event marked by the collision of the entities
(4) Information for damage determination. This information, when available, shall be used by each entity to determine the extent of damage received during the collision. This information includes:
   (a) The velocity vector of the issuing entity
   (b) The mass of the issuing entity
   (c) The location of impact in entity coordinates of the entity with which the issuing entity collided

   These fields shall all be used or all be zero.

4.4.5.1.2 Issuance of the Collision PDU. The Collision PDU shall be issued by an entity when a collision is detected between the issuing entity and an object or some other entity taking part in the simulation exercise. If the collision involves two entities, both entities shall issue the Collision PDU even if only one of them detected the collision. An entity that receives a Collision PDU without detecting such a collision shall issue a Collision PDU to the entity that issued the first Collision PDU.

The Collision PDU shall be issued by using a real-time, reliable, multicast communication service.

4.4.5.1.3 Receipt of the Collision PDU. Upon receipt of the Collision PDU, the data contained therein shall be used to record the event and to determine the extent of the damage sustained in the collision.

4.4.6 Simulation Management PDUs for DIS. The following paragraphs shall establish the content and the procedure for use of Simulation Management PDUs in a DIS exercise.

4.4.6.1 Simulation Management PDU Header. Certain information shall be included with each Simulation Management PDU. This information includes: standard PDU header (see 4.4.1), originating entity (entity issuing the PDU), and receiving entity.

4.4.6.1.1 PDU Header. This field shall be required at the beginning of each DIS PDU. It shall contain information about the protocol version, the exercise identification and the type of PDU. PDU types for Simulation Management are:

(1) Create Entity
(2) Remove Entity
(3) Start/Resume
(4) Stop/Freeze
(5) Acknowledge
(6) Action Request
(7) Action Response
(8) Data Query
(9) Set Data
(10) Data
(11) Event Report
(12) Message

4.4.6.1.2 Entity IDs. Identification numbers assigned to specific entities participating in a DIS exercise shall not have a value of zero or all ones. These values are reserved for other designations.

4.4.6.1.2.1 Originating Entity ID. The identification for the entity responsible for issuing the PDU shall be specified. This field shall be composed of Site ID, Application ID, Entity ID and Group ID.

4.4.6.1.2.2 Receiving Entity ID. The identification of the intended receiving entity shall be specified. This field shall be composed of Site ID, Application ID, Entity ID and Group ID. To broadcast this PDU to all entities, the receiving entity ID shall have a value of all ones (binary). If the PDU is a Create Entity PDU, the Receiving Entity ID field shall represent the entity ID assigned to a new entity if it is known.

If the receiving simulation application is intended to assign the entity ID to the newly created entity, then the value of the Entity ID field in the Create Entity PDU shall be the maximum value for this field minus one ($2^{16}-2$). In any case, the Entity ID of the new entity is returned in Originating Entity ID field of the Acknowledge PDU.

4.4.6.2 The Simulation Management Computer. Simulation Management may be performed by any computer on the DIS network. The computer may be dedicated to the task of simulation management or it may be another simulator or interface to a simulator, or it may be the same computer that is simulating the entity (running the simulation application). An exercise may have one Simulation Manager, or multiple Simulation Managers. One entity may interact with multiple Simulation Managers during an exercise. In all cases, the computer that is serving as Simulation Manager or is performing simulation management functions shall be referred to in this document as the Simulation Manager (SM).

4.4.6.3 Simulation Management Functions. The Simulation Management functions defined in this document serve to establish a portion of the Session Database[2] for simulations participating in a DIS Exercise. These functions can be categorized as entity/exercise management and data management. Many actions that are performed with individual entities can also be performed exercise wide. For example, an entity or an exercise can be initialized, started, or stopped. Many entity/exercise actions are, therefore, explained on the entity level with the understanding that the same actions may also apply on an exercise level as well. Information associated with performing these functions is communicated by a number of different PDUs.

The PDUs required for simulation management are described in 4.4.6.4 and the subparagraphs that follow. Section 4.4.6.5 discusses the simulation management functions and how the PDUs shall be used to accomplish them.

4.4.6.4 Simulation Management PDUs. The following PDUs shall be used to perform simulation management functions:

4.4.6.4.1 Create Entity PDU. The Create Entity PDU shall communicate information about the creation of a new entity for a DIS exercise. This PDU simply establishes the identity of the new entity.

4.4.6.4.1.1 Information Contained in the Create Entity PDU. The Create Entity PDU shall contain the same information found in the Simulation Management PDU header. The Originating Entity ID shall represent the ID number for the SM that is responsible for creating the new entity. The Receiving Entity ID shall represent the Entity ID number of the newly created entity if it is known.

---

[2]The Session Database is defined in the DIS Strawman Architecture document as "a standard database which includes network initialization data and simulation entity initialization and control data." The scope of the Simulation Management PDUs is currently entity initialization and control.

4.4.6.4.1.2 Issuance of the Create Entity PDU. The Create Entity PDU shall be issued by a SM when a new entity is to be created for a DIS exercise. This PDU is not necessary for the creation of munition entities for which tracking information is necessary.

The Create Entity PDU shall be issued using a best-effort, unicast or multicast communication service.

4.4.6.4.1.3 Receipt of the Create Entity PDU. Upon receipt of the Create Entity PDU, if the receiving simulation application can comply with the Create Entity PDU request, then it shall assign an entity ID value to the new entity shall assign the entity ID value to a new entity. The receiving simulation application shall respond to the Create Entity PDU by issuing an Acknowledge PDU with the Response Flag field indicating that it can comply with the request, or why it cannot.

4.4.6.4.2 Remove Entity PDU. The Remove Entity PDU shall communicate the removal of an entity from a DIS exercise. This PDU indicates to the receiving entity that it is being removed from the exercise.

4.4.6.4.2.1 Information Contained in the Remove Entity PDU. The Remove Entity PDU shall contain the same information found in the SM PDU header. The Originating Entity ID shall represent the ID number for the SM issuing the Remove Entity PDU. The Receiving Entity ID shall represent the entity that is being removed from the simulation exercise.

4.4.6.4.2.2 Issuance of the Remove Entity PDU. The Remove Entity PDU shall be issued by an SM when a particular entity is to be removed from the simulation.

The Remove Entity PDU shall be issued using a best-effort, unicast or multicast communication service.

4.4.6.4.2.3 Receipt of the Remove Entity PDU. Upon receipt of the Remove Entity PDU, the receiving simulation application shall immediately cease simulating its entity and remove it from the simulation exercise as quick as possible. The receiving simulation application shall then acknowledge the receipt of the Remove Entity PDU by issuing an Acknowledge PDU.

4.4.6.4.3 Start/Resume PDU. The Start/Resume PDU shall be used to communicate to a simulation entity that it is to leave a stopped/Frozen state and begin participating in a simulation exercise. This PDU indicates to the entity the time it is to join the simulation.

4.4.6.4.3.1 Information Contained in the Start/Resume PDU. The Start/Resume PDU shall contain the following information:

(1) Standard Simulation Management header information

(2) The time of day in real world time with respect to Greenwich time

(3) The time of day in the simulation exercise

4.4.6.4.3.2 Issuance of the Start/Resume PDU. The Start/Resume PDU shall be issued by an SM to an entity to instruct that entity to proceed from a stopped/Frozen state to a simulated state (see 4.4.6.5.3 for more on entity states). The Start/Resume PDU must be sent far enough in advance that any receiving simulation application will have time to comply. The maximum expected time between transmission and reception of PDUs under various conditions is described in document IST-CR-93-07.

The Start/Resume PDU shall be issued using a best effort, unicast or multicast communication service.

4.4.6.4.3.3 Receipt of the Start/Resume PDU. Upon receipt of the Start/Resume PDU, the receiving entity shall respond by issuing an Acknowledge PDU. The receiving entity shall leave the stopped/frozen state and join the exercise at the real world time indicated in the PDU. The entity shall simulate the time of day indicated in the Simulation Time field.

4.4.6.4.4 Stop/Freeze PDU. The Stop/Freeze PDU shall be used by the SM to indicate to a simulated entity that it shall leave a simulating state and enter a stopped state. This PDU shall also indicate the reason that the receiving entity was requested to stop. The Stop/Freeze PDU must be sent far enough in advance that any receiving simulation application will have time to comply. The maximum expected time between transmission and reception of PDUs under various conditions is described in document IST-CR-93-07.

4.4.6.4.4.1 Information Contained in the Stop/Freeze PDU. The Stop/Freeze PDU shall contain the following information:

(1) Standard Simulation Management header information

(2) The reason the SM requested that the entity stop simulating

4.4.6.4.4.2 Issuance of the Stop/Freeze PDU. The Stop/Freeze PDU shall be issued by an SM to an entity when the SM requests an entity to stop simulating.

The Stop/Freeze PDU shall be issued using a best effort, unicast or multicast communication service.

4.4.6.4.4.3 Receipt of the Stop/Freeze PDU. Upon receipt of the Stop/Freeze PDU, the receiving simulator shall respond by issuing an Acknowledge PDU. The receiving entity shall leave the simulating state at the real world time indicated in the PDU. The stop/freeze may be temporary or indefinite based on the reason for stop/freeze.

4.4.6.4.5 Acknowledge PDU. The Acknowledge PDU shall be used to acknowledge the receipt of a Create Entity PDU, a Remove Entity PDU, a Start/Resume PDU, or a Stop/Freeze PDU. This PDU verifies to the SM the receipt of the issued PDU. The acknowledge PDU shall be sent as soon as possible after the receipt of the above PDUs.

4.4.6.4.5.1 Information Contained in the Acknowledge PDU. The Acknowledge PDU shall contain the following information:

(1) Standard Simulation Management header information. The Originating Entity ID refers to the entity issuing the Acknowledge PDU. The Receiving Entity ID shall refer to the SM to whom the entity is responding.

(2) A flag to indicate the type of PDU message to which the acknowledgement pertains.

4.4.6.4.5.2 Issuance of the Acknowledge PDU. The Acknowledge PDU shall be issued by a simulation application in response to a Create Entity PDU, a Remove Entity PDU, a Start/Resume PDU, or a Stop/Freeze PDU.

The Acknowledge PDU shall be issued using a real-time, best-effort, unicast or multicast communication services.

4.4.6.4.5.3 Receipt of the Acknowledge PDU. Upon receipt of the Acknowledge PDU, the receiving SM shall note that the simulation application has successfully received the previous PDU transmission.

4.4.6.4.6 Action Request PDU. The Action Request PDU shall be used by the SM to request that a specific action be performed by a simulation entity. Information required for the entity to perform the requested action shall also be included in this PDU.

4.4.6.4.6.1 Information Contained in the Action Request PDU. The Action Request PDU shall contain the following information:

(1) Standard Simulation Management header information

(2) A counter for tracking requests

(3) Identification of the specific action to be taken

(4) Data required for performing the requested action

4.4.6.4.6.2 Issuance of the Action Request PDU. The Action Request PDU shall be issued by an SM when the SM requires that an entity perform a particular action.

The Action Request PDU shall be issued using a best effort, unicast or multicast communication service.

4.4.6.4.6.3 Receipt of the Action Request PDU. Upon receipt of the Action Request PDU, the receiving entity shall acknowledge the receipt by issuing an Action Response PDU. The Action Response PDU shall be used by the receiving entity to indicate the action taken as a result of receipt of the Action Request PDU.

4.4.6.4.7 Action Response PDU. The Action Response PDU shall be used by an entity to acknowledge the receipt of an Action Request PDU. This PDU shall provide information on the status of the request and may also be used to provide additional information depending upon the type of Action requested.

4.4.6.4.7.1 Information Contained in the Action Response PDU. The Action Response PDU shall contain the following information:

(1) Standard Simulation Management header information

(2) Request identification number for the action being processed

(3) Status of the action request

(4) Datum values which may be requested as a by-product of the action request

4.4.6.4.7.2 Issuance of the Action Response PDU. The Action Response PDU shall be issued upon receipt of an Action Request PDU.

The Action Response PDU shall be issued using a best-effort, unicast or multicast communication service.

4.4.6.4.7.3 Receipt of the Action Response PDU. Upon receipt of the Action Response PDU, the receiving entity (originator of the Action Request PDU) shall note that the Action Request was received and the status of that request.

4.4.6.4.8 Data Query **PDU.** The Data Query PDU shall be used by an SM to communicate a request for data from a simulated entity.

4.4.6.4.8.1 Information Contained in the Data Query PDU. The Data Query PDU shall contain the following information:

(1) Standard Simulation Management header information

(2) A time interval between issues of requested information at regular basis

(3) The quantity and types of data requested

4.4.6.4.8.2 Issuance of the Data Query PDU. The Data Query PDU shall be issued by an SM when the SM requests data from the receiving entity.

The Data Query PDU shall be issued using a best effort, unicast or multicast communication service.

4.4.6.4.8.3 Receipt of the Data Query PDU. Upon receipt of the Data Query PDU, the receiving entity shall respond by issuing a Data PDU containing the requested data.

4.4.6.4.9 Set Data PDU. The Set Data PDU shall be used by the SM to set or change certain parameters in an entity.

4.4.6.4.9.1 Information Contained in the Set Data PDU. The Set Data PDU shall contain the following information:

(1) Standard Simulation Management header information

(2) Number of fixed datum, the datum IDs and the specific datum values

(3) Number of Variable datum, the datum IDs, the datum length and the specific datum values

4.4.6.4.9.2 Issuance of the Set Data PDU. The Set Data PDU shall be issued by SM when it is necessary to set or change parameters of an entity.

The Set Data PDU shall be issued using a best effort, unicast or multicast communication service.

4.4.6.4.9.3 Receipt of the Set Data PDU. Upon receipt of the Set Data PDU, the receiving entity shall set the appropriate parameters as specified in the Set Data PDU. It shall be up to the receiving entity of the Set Data PDU to determine which (if any) parameters described in the Set Data PDU it can set. The receiving entity shall then respond with a Data PDU. The Data PDU shall verify the receipt of the Set Data PDU by returning the parameter values that were set in response to the Set Data PDU. Parameters that were set to the same values as in the Set Data PDU shall be set to those values in the Data PDU. Parameter values that were set to different values than requested shall be set to their actual values in the Data PDU. Parameters to which the receiving entity cannot comply shall not be included in the Data PDU response.

4.4.6.4.10 Data PDU. The Data PDU shall be used by an entity in response to a Data Query PDU or a Set Data PDU. This PDU allows the entity to provide requested information in a Data Query PDU. It also allows the entity to confirm the information received in a Set Data PDU.

4.4.6.4.10.1 Information Contained in the Data PDU. The Data PDU shall contain the following information:

(1) Standard Simulation Management header information

(2) Number of fixed datum, the datum IDs and the specific datum values

(3) Number of Variable datum, the datum IDs, the datum length and the specific datum values

4.4.6.4.10.2 Issuance of the Data PDU. The Data PDU shall be issued by an entity in response to a Data Query PDU. In this case, the Data PDU shall contain the information requested in the Data Query PDU. The Data PDU shall also be used to respond to the Set Data PDU. In response to the Set Data PDU, the Data PDU shall contain the same information sent to the entity in the Set Data PDU. This response shall be a confirmation that specified parameters have been set as designated in the Set Data PDU.

4.4.6.4.10.3 Receipt of the Data PDU. Upon receipt of the Data PDU, the simulation management computer shall record the information for simulation management purposes.

The Data PDU shall be issued using a best-effort, unicast or multicast communication service.

4.4.6.4.11 Event Report PDU. The Event Report PDU shall be used to communicate the occurrence of a significant event. What constitutes a significant event may have been set previously by the SM, or it may be internal to the simulation application.

4.4.6.4.11.1 Information Contained in the Event Report PDU. The Event Report PDU shall contain the following information:

(1) Standard Simulation Management header information

(2) Identification of the type of event that has occurred

(3) Datum values that may be relevant to the reporting of the event

4.4.6.4.11.2 Issuance of the Event Report PDU. The Event Report PDU shall be issued by an entity to an SM to communicate that a significant event has occurred. This PDU shall be directed to the SM that designated the significant event to be reported.

The Event Report PDU shall be issued using a best-effort, unicast or multicast communication service.

4.4.6.4.11.3 Receipt of the Event Report PDU. Upon receipt of the Event Report PDU, the SM shall record the event for later use.

4.4.6.4.12 Message PDU. The Message PDU shall be used to input a message into a data stream either for use as a comment, error or test message, or as a place holder in a sequentially stored exercise. The Message PDU may also be used for other message needs in the simulation exercise.

4.4.6.4.12.1 Information Contained in the Message PDU. The Message PDU shall contain the following information:

(1) Standard Simulation Management header information

(2) The number of fields for communicating the message and the specific message information

4.4.6.4.12.2 Issuance of the Message PDU. The Message PDU shall be issued for storing messages into the data stream. The message may be addressed to a specific entity, to all entities, or to no specific entity. The Receiving Entity field shall contain a value of all zeros if the message is addressed to no specific entity and all ones (binary) if addressed to all entities.

4.4.6.4.12.3 Receipt of the Message PDU. Upon receipt of the Message PDU, the receiving simulator shall record the message for later use.

The message PDU shall be issued using a best-effort, unicast or multicast communication service.

4.4.6.5 Entity/Exercise Management. Management of an entity or exercise includes the capability to create new entities, initialize or change entity or exercise parameters, start or stop an entity or exercise, request an entity to perform a specific action and record significant entity data or exercise events. Each of these actions are described in the paragraphs that follow.

4.4.6.5.1 Entity Creation. The Simulation Management Protocol provides three ways to create a new entity. The first method allows the SM to establish the ID (using Create Entity PDU) of the new entity, query for data about the new entity (using Data Query PDU), and set initial parameters for the new entity (using Set Data PDU). The second method is similar to the first method except the SM does not query for data. The third method is even more streamlined, requiring only the Create Entity PDU. It should be noted that the third method relies upon certain database information (session database) to be established in advance of the exercise start, whereas the first method allows the entire creation and initialization process to proceed with little information established in advance. The three methods of entity creation are described in the sub-paragraphs below.

4.4.6.5.1.1 Entity Creation, Query and Initialization. To create, query and initialize a new entity, either for an existing exercise or for a new exercise, the SM shall begin by issuing a Create Entity PDU to the simulation application that will be controlling the simulation entity. The receiving simulation application shall respond with an Acknowledge PDU. These actions simply assign an ID number to a new entity.

The SM then shall request that certain data be issued by the simulation application controlling the new entity. This is accomplished by issuing a Data Query PDU. The simulation application shall respond by sending the requested data using a Data PDU.

The SM then shall send necessary initialization information to the new entity using a Set Data PDU. The receipt of the Set Data PDU by the entity shall be indicated by the return of a Data PDU.

The process of entity creation, query and initialization is illustrated in Figure 4-5.1.

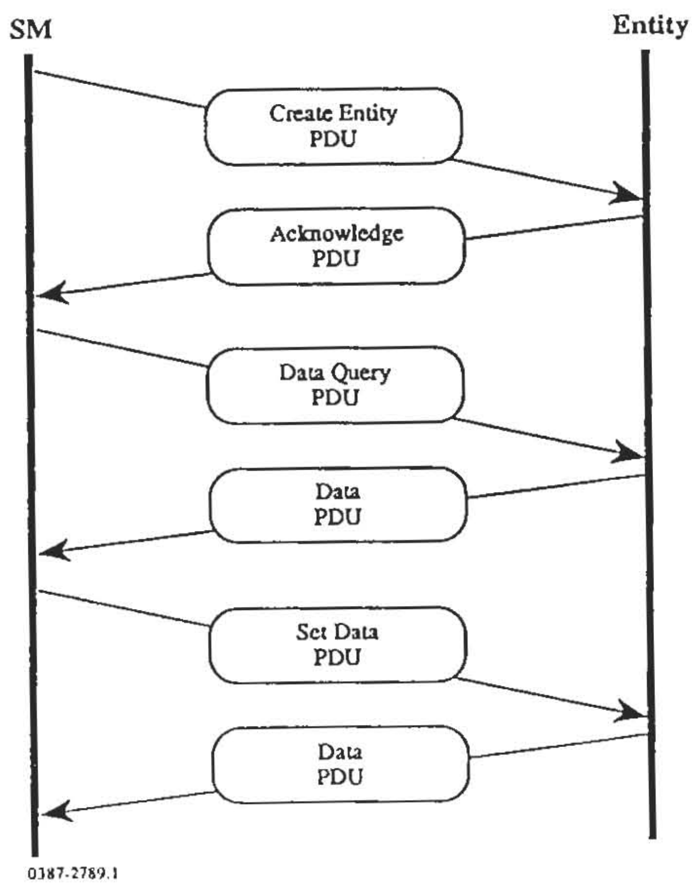


Figure 4-5.1
Entity Creation, Query and Initialization

4.4.6.5.1.2 Entity Creation and Initialization. To create and initialize a new entity, the SM shall begin by issuing a Create Entity PDU to the simulation application that will be controlling the simulation entity. The receiving simulation application shall respond with an Acknowledge PDU. These actions simply assign an ID number to a new entity.

It is assumed that the SM has all the necessary data it needs for the new entity (type of entity and characteristics). This information shall be established off-line and prior to the entity creation. The SM then shall initialize the new entity by issuing Set Data PDU to the simulation application controlling the new entity. The receipt of the Set Data PDU by the entity shall be indicated by the return of a Data PDU.

The process of entity creation and initialization is illustrated in Figure 4-5.2.

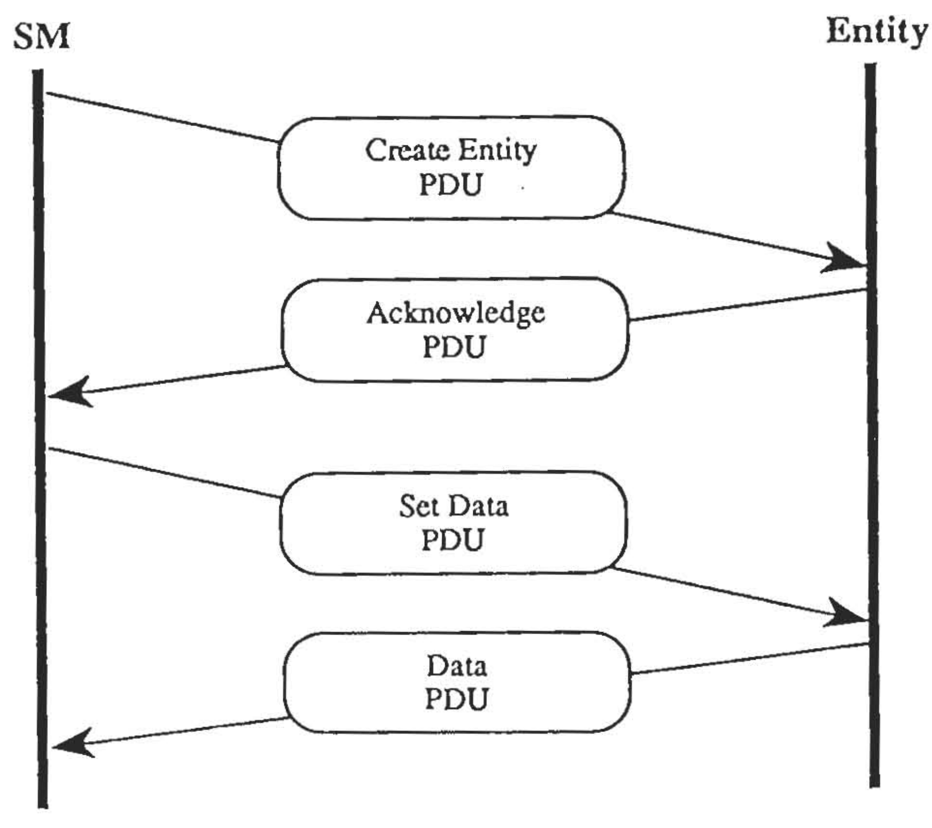


Figure 4-5.2
Entity Creation and Initialization

4.4.6.5.1.3 Entity Creation. In the case that necessary entity data and initialization data have already been established off-line and prior to the exercise, it is possible to create a new entity by assigning a particular entity ID. To create a new entity, the SM shall issue a Create Entity PDU to the simulation application that will be controlling the simulation entity. The receiving simulation application shall respond with an Acknowledge PDU.

The process of entity creation is illustrated in Figure 4-5.3.

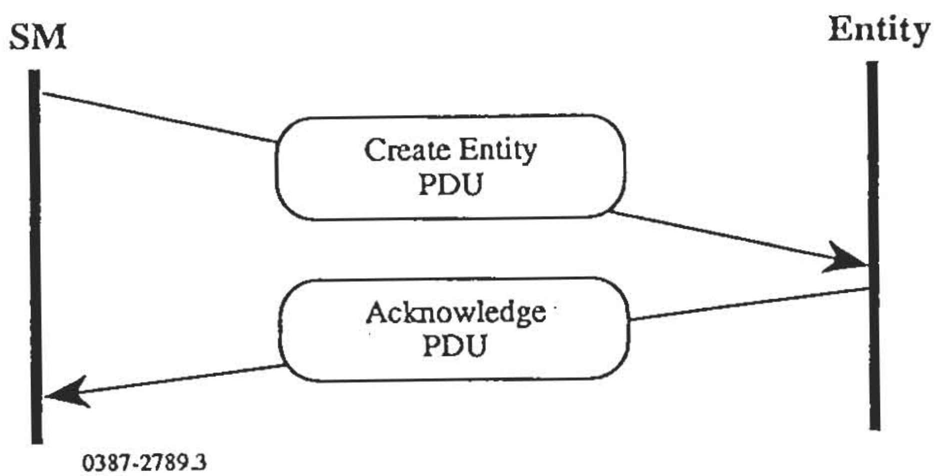


Figure 4-5.3
Entity Creation

4.4.6.5.2 Changing Entity Parameters. Parameters within a particular entity shall be changed by the SM by the issue of a Set Data PDU. For example, during initialization, the Set Data PDU requests that an entity sets (or changes) certain parameters of its internal state to specified values. As in initialization, the receiving entity shall respond to the Set Data PDU by issuing a Data PDU. The issued Data PDU shall contain the same information as the Set Data PDU. This will serve as an acknowledgement to the SM that the correct changes were made. Another Set Data PDU shall be issued if the first was incorrectly received.

4.4.6.5.3 Starting or Stopping an Entity. An entity shall be in one of three states: Ready, Stopping or Simulating. The entity may not exist in the simulation although it is ready to be initialized. In this case, the entity is in the Removed state. The entity may also be in the Stopped state when it is not simulating, but is able to be started at any time. Finally, an entity may be in the Simulating state. These states are represented in Figure 4-6.

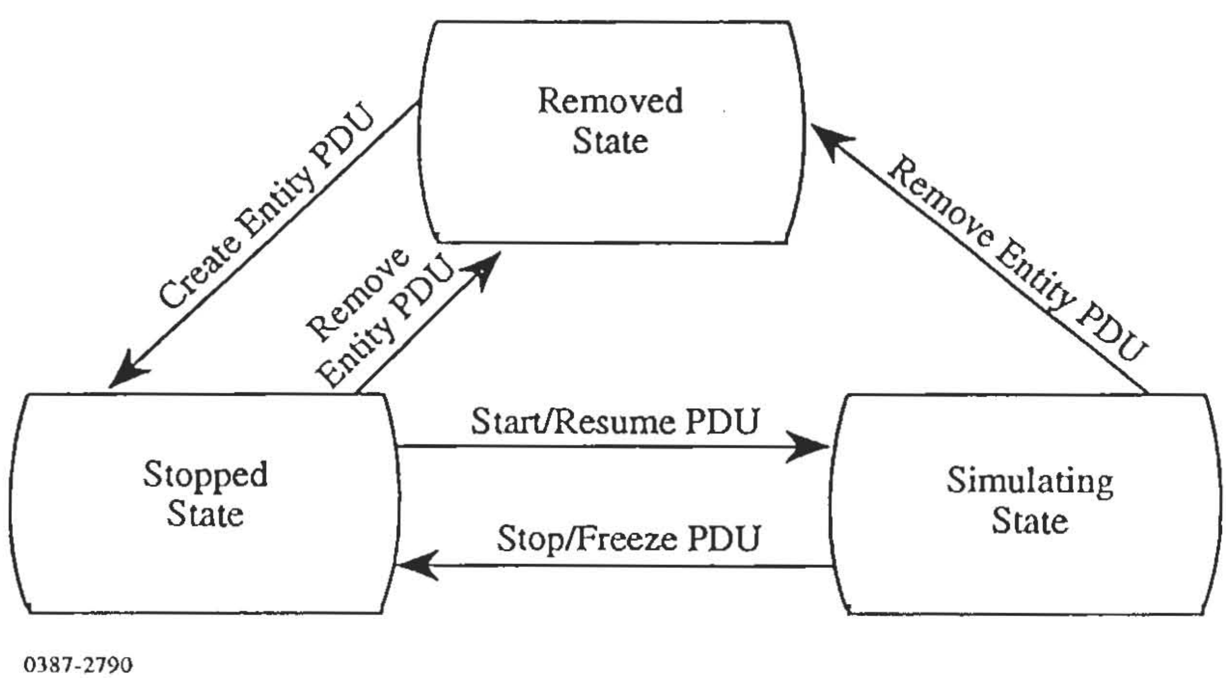


Figure 4-6
Entity States in Simulation Management

An entity is in the Removed state until it is created and initialized (see 4.4.6.5.1). At this point the entity is in the Stopped/Frozen State. To Start an entity, an SM shall issue a Start/Resume PDU to the entity to be started. The receiving entity shall respond with an Acknowledge PDU. Similarly, to Stop/Freeze a simulating entity, an SM shall issue a Stop/Freeze PDU to the entity to be stopped. The receiving entity shall likewise respond with an Acknowledge PDU. Figure 4-7 illustrates this action.

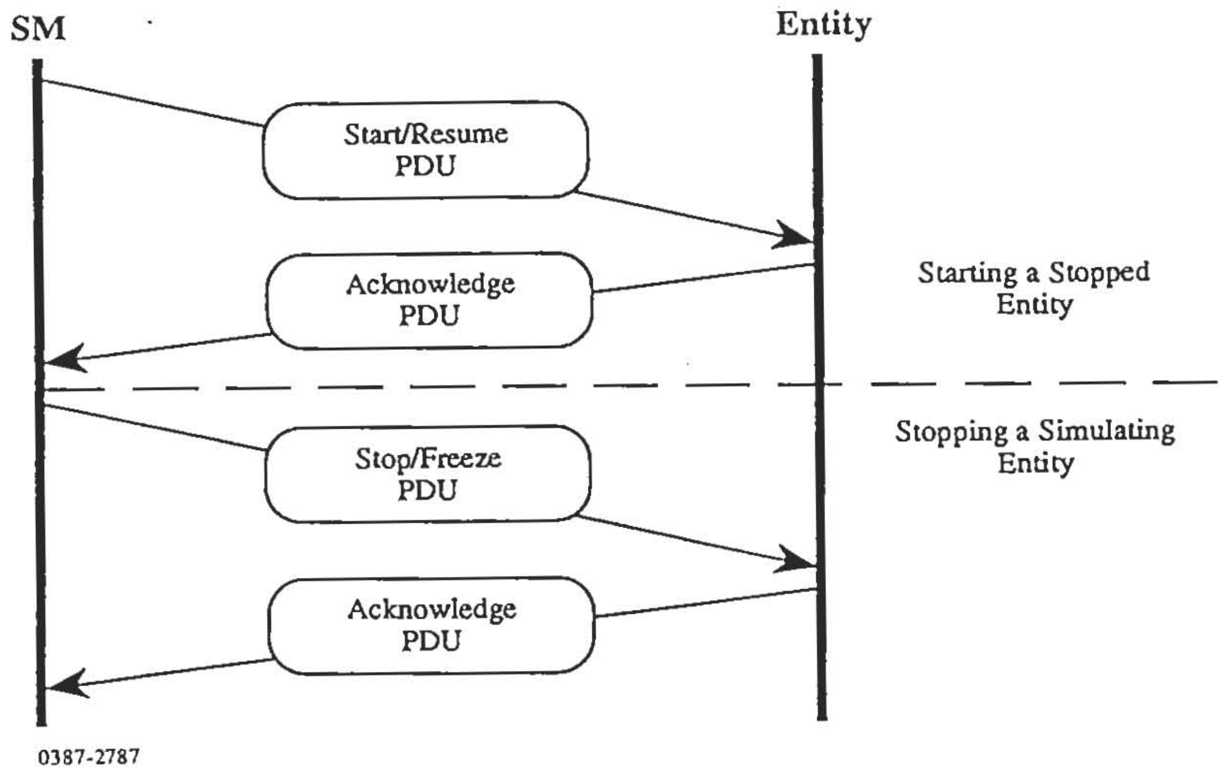


Figure 4-7
Starting/Stopping an Entity

4.4.6.5.4 Removing an Entity From an Exercise. An entity shall be removed from an exercise by an SM. To remove the entity, an SM shall issue a Remove Entity PDU to the entity to be removed. The simulation application controlling the entity shall respond with an Acknowledge PDU and shall cease simulating its entity. These actions are represented in Figure 4-8.

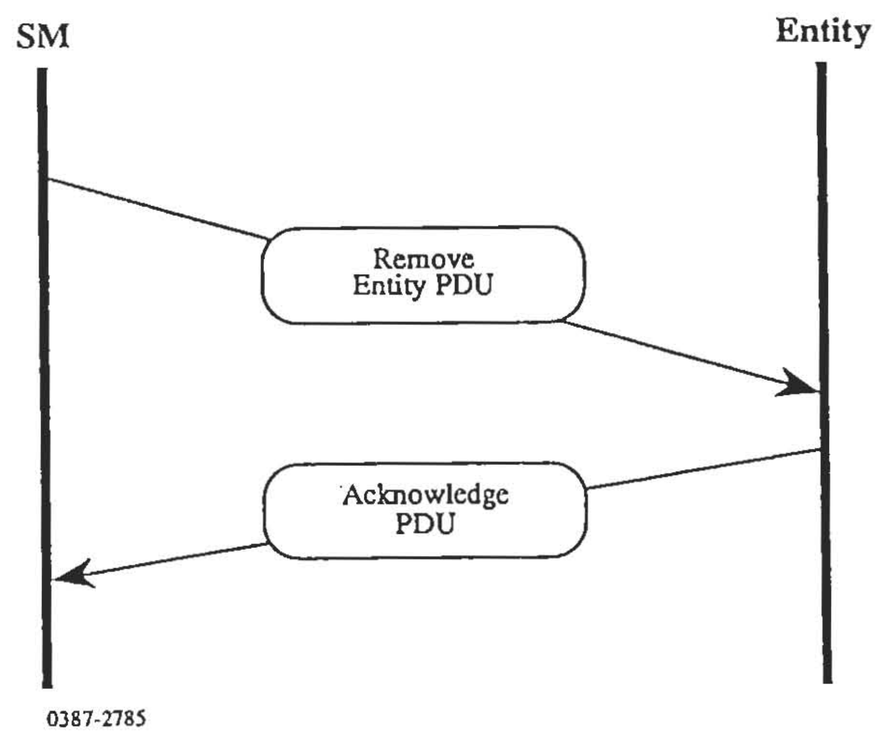


Figure 4-8
Removing an Entity

4.4.6.5.4 Requesting an Entity to Perform an Action. An entity may be requested to perform a specific action by the SM. In addition to requesting the action, the SM may provide needed information for performance of the requested action.

To request an action, the SM shall issue an Action Request PDU to the entity to perform the action. Upon receipt of the Action Request PDU, the receiving entity shall act upon the request and respond with an Action Response PDU. This interaction is represented in Figure 4-9.

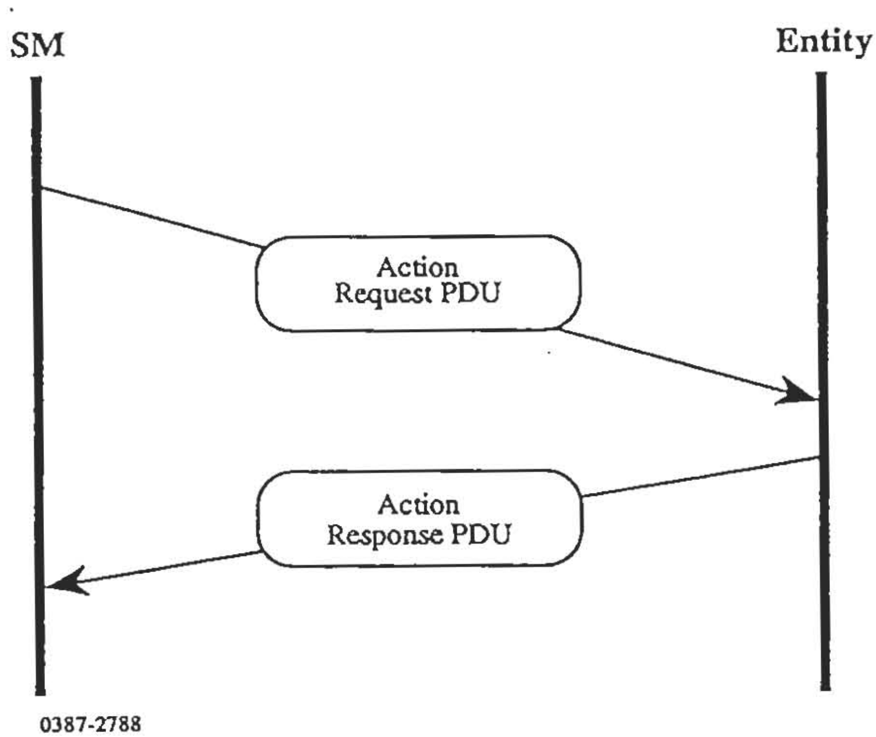


Figure 4-9
Action Request/Response

4.4.6.5.5 Event Recording. To keep record of certain key events that may occur during the course of a DIS exercise, an Event Report PDU shall be used in the reporting of such events. The SM may set certain parameters such that a particular event is considered key. When that event occurs, the entity involved issues an Event Report PDU to report the event.

4.4.6.6 Data Management. In addition to managing entities and the exercise, data management may be accomplished using the simulation management PDUs.

4.4.6.6.1 Request for Data. An SM shall request data concerning the internal state of an entity by issuing a Data Query PDU. The receiving entity shall respond by providing the requested data in a Data PDU. These actions are represented in Figure 4-10.

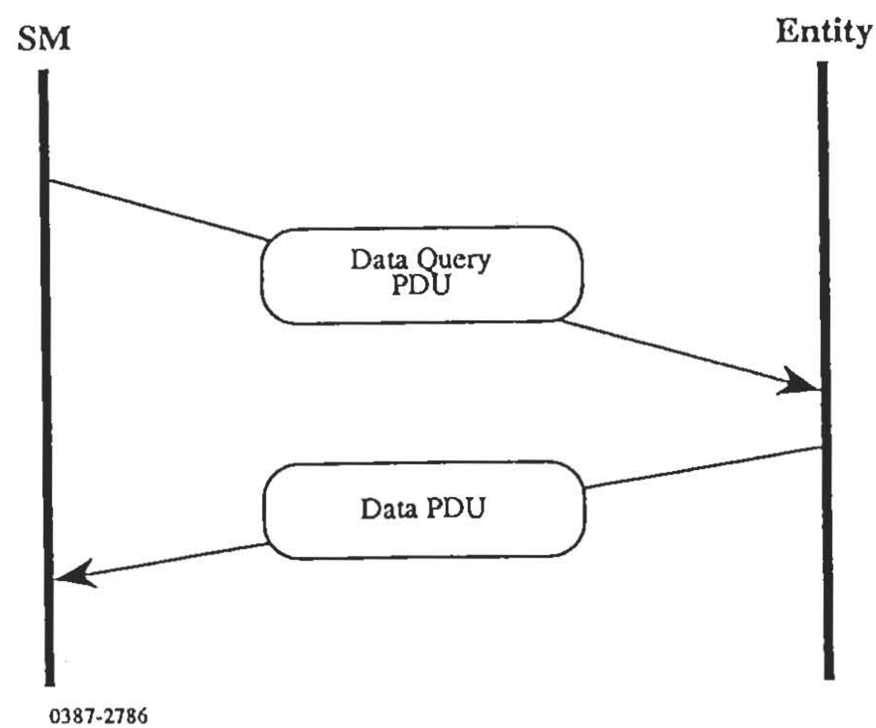


Figure 4-10
Requesting Entity Data

4.4.6.6.2 Setting or Changing Internal State Values. Internal state information can be set or modified by the SM. This shall be accomplished by the SM issuing a Set Data PDU indicating the Data to be changed or set. It shall be up to the receiving entity of the Set Data PDU to determine which (if any) parameters described in the Set Data PDU it can set. The receiving entity shall respond by issuing a Data PDU. The Data PDU shall verify the receipt of the Set Data PDU by returning the parameter values that were set in response to the Set Data PDU. Parameters that were set to the same values as in the Set Data PDU shall be set to those values in the Data PDU. Parameter values that were set to different values than requested shall be set to their actual values in the Data PDU. Parameters to which the receiving entity cannot comply shall not be included in the Data PDU response. This serves as an acknowledgement to the SM. If the SM determines that the original request was incorrectly received, another Set Data PDU may be issued. These actions are represented in Figure 4-11.

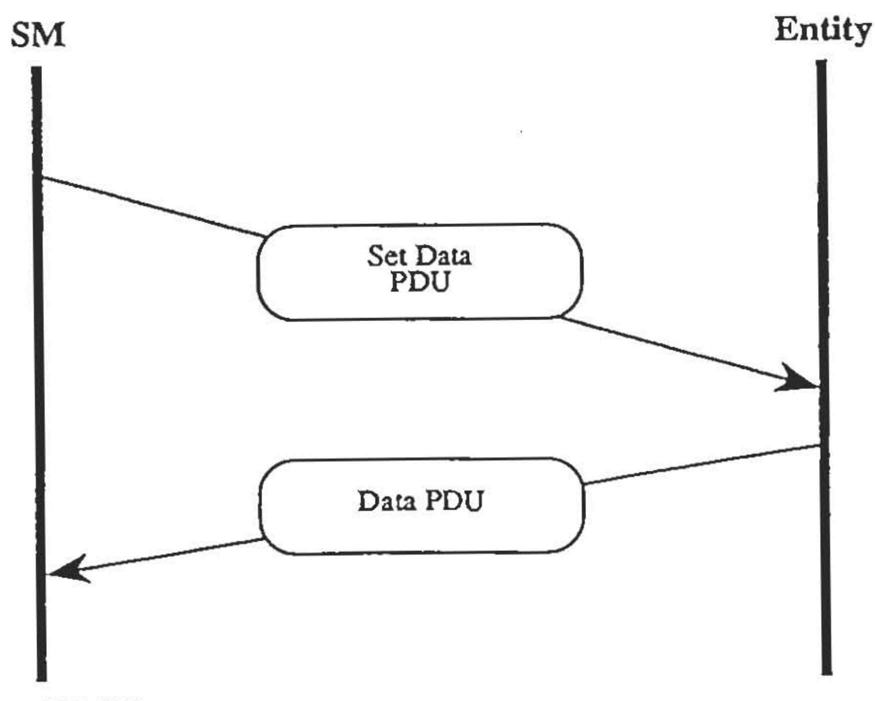


Figure 4-11
Setting/Changing Internal State Values

4.4.6.6.3 Entity Reconstitution. A killed or damaged entity shall allow its reconstitution by an SM when requested. To reconstitute an entity, an SM shall issue a Stop/Freeze PDU. The receiving entity shall respond with an Acknowledge PDU. The entity shall not be removed but shall be in the Stopped state as in Figure 4-6. The SM shall then issue a Set Data PDU to reset the entity parameters. The receiving entity shall respond with a Data PDU. The SM shall then issue a Start/Resume PDU for the receiving entity to re-join the exercise. The receiving entity shall respond with an Acknowledge PDU. These actions are represented in Figure 4-12.

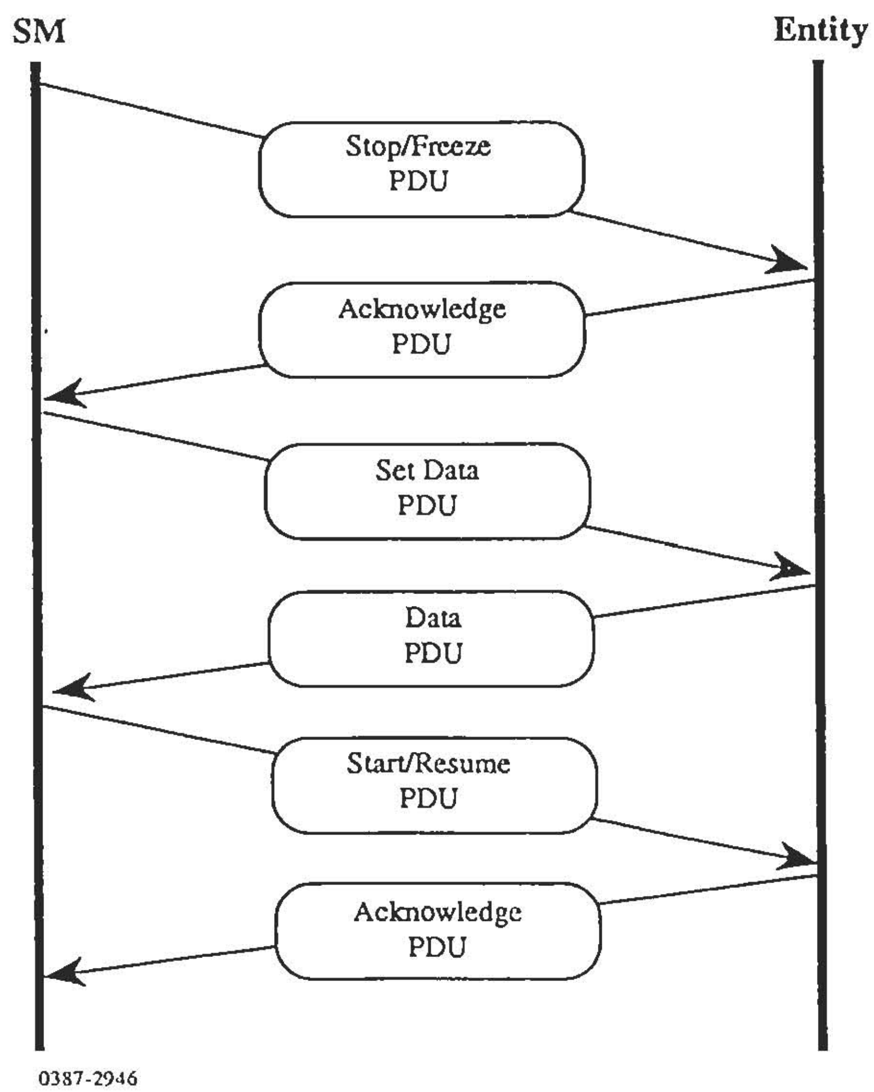


Figure 4-12
Reconstituting an Entity

4.4.7 Distributed Emission Regeneration for DIS. The following paragraphs shall establish the content and procedure for the use of the Distributed Emission Regeneration (DER) in a DIS exercise.

4.4.7.1 Distributed Emission Regeneration Approach. The following assumptions shall be followed for DER in DIS:

(1) Entities with emitters shall simulate their emitter and shall output predefined, real-time operational parameters via the DIS communications network.

(2) Entities with receivers shall regenerate the transmitter signal to the fidelity level required by that particular receiving simulation application. Regeneration shall be accomplished by using the operational parameters provided in the DER PDU along with information from stored databases which describe the transmitter capabilities (i.e. beam patterns, etc.).

(3) Scan patterns shall be regenerated based on center beam data sent from the transmitter coupled with receiver stored database parameters.

(4) The DER approach allows but does not require transmitting entities to communicate information on entities which are in the entity's tracking beam. It should be noted that when this information is provided in the Emission PDU, it does not indicate whether or not target is being tracked by the sending entity.

4.4.7.2 Emission PDU. The Emission PDU shall be used to communicate active EW and acoustic emissions, including active countermeasures.

4.4.7.2.1 Information Contained in the Emission PDU. The Emission PDU shall contain the following information:

(1) Standard PDU Header information.

(2) Identification of the emitting entity.

(3) Identification of the event.

(4) Identification of the type of update information in PDU.

A state update provides a full description of the emission system(s) identified in the PDU. Only active (emitting) systems and active beams will be included in the state update. All

A change data update provides a method to allow changes of emitter state to be communicated between state updates.

(5) Number of emitter systems for which information is being provided in the PDU.

(6) Information for one or more emission systems that the entity has. This information includes:

(b) Length of the system data.

(c) Number of beams (for each system) for which information is being provided in the PDU.

(d) Emitter System which includes the emitter name, function of the emission system and emitter id number.

(e) Location of the emitter which is the location of the antenna beam source.

(f) Information for one or more beams that the system has. This information includes:

(i) Length of the beam data.

(ii) Beam id # for each beam.

(iii) Beam parameter index.

(iv) Fundamental parametric data which is essentially data that can vary for a specific system or can vary dynamically during system operation (even though this system's mode and beam functions are not changed). This data is also available to support applications of low-fidelity simulations which may not have the computational power to process high-fidelity regeneration models.

(v) Beam function identifier.

(vi) Number of targets for which information is provided in the track/jam field of the PDU for the beam being defined.

(vii) High density track/jam field.

(viii) Jamming mode sequence to define active jamming techniques being applied.

(ix) Information for one or more targets that is in track/jam field. This information includes:

1

- Track/jam field which provides target site, host and entity id's for targets which the source entity identifies as being in the illumination area of a track beam or, in the case of a jamming emission, provides the site, host, entity, emitter and beam id upon which the jamming emission is acting.

4.4.7.2.2 Issuance of the Emission PDU. The Emission PDU shall be issued in the following instances:

(1) Operational parametrics for the emitter change.

(2) Center Beam Descriptors exceed specified thresholds. Unless otherwise specified by the user, the default threshold azimuth and elevation shall be one degree.

(3) Entities enter or leave the emitter's track beam.

(4) Providing a state update at predetermined intervals of time (5 seconds).

The Emission PDU shall be issued using a real-time, best effort, multicast communication service.

4.4.7.2.3 Receipt of the Emission PDU. Upon receipt of an Emission PDU, the receiving entity 2 shall determine if the emission is detectable and use the information in the Emission PDU to appropriately influence emission detection equipment in the simulation.

4.4.7.2.4 Emission Regeneration. Emissions shall be generated based on parametrics stored in databases of the receiving entity (as pointed to by the emission PDU emitter name and beam parameter index field(s) and based on the dynamic fundamental parameter data passed in the emission PDU.

4.4.7.3 Laser PDU. The Laser PDU shall be used to communicate information for lasing functions in support of a laser-guided weapon engagement.

4.4.7.3.1 Information Contained in a Laser PDU. The Laser PDU shall contain the following information:

(1) Standard PDU header

(2) Identification of the entity performing the lasing

(3) Code Name for the laser system

(4) Time of the lasing information

(5) Identification of the entity being lased (provided only if laser spot is on an entity)

(6) Code of the laser

(7) Output power of the laser

(8) The laser wavelength

(9) Laser spot position with respect to an entity. This data provides specific detail of the spot position with respect to an entity's center of gravity.

(10) Location of the laser spot

4.4.7.3.2 Issuance of the Laser PDU. The Laser PDU shall be issued at a rate of 10 Hz whenever a laser is active. When the laser goes inactive, an additional Laser PDU shall be issued with the laser power field set to zero.

The Laser PDU shall be issued using a real-time, best-effort, multicast communication service.

4.4.7.3.3 Receipt of the Laser PDU. Upon receipt of the Laser PDU, the entity firing the laser-guided munitions shall use the information contained in the PDU to simulate the guidance and the final detonation of the weapon after it is fired. For the guidance and detonation simulations the entity firing the weapon shall use PDU information from the field defining the location of the laser spot. At the time of an impact the firing entity will check to see if there is information in the PDU field for identification of a lased entity (for accuracy a correlation should be made between the impact time and the PDU timestamp). If there is no entity identified as being lased then the firing entity shall output the laser spot field information as part of the detonation PDU. If there is an entity identified as being lased then the firing entity shall use the laser spot with respect to an entity and information from that target's entity state PDU to provide an accurate impact position for the detonation PDU. The laser spot position with respect to an entity shall also be used in the Detonation PDU for the location with respect to entity field.

4.4.7.4 Expendables PDU: TBD

4.4.7.5 IFF PDU: TBD

4.4.8 Radio Communications Protocol. The following paragraphs define the protocol for the simulation of radio communications in DIS. The Radio Communications Protocol (RCP) supports the simulation of both audio and data transmission by radio. In the RCP, the content of the radio transmission may be conveyed in its entirety in real-time, or may be conveyed by reference to a prerecorded database.

4.4.8.1 Radio Communications Approach. The following assumptions shall be followed for radio communications in DIS:

(1) Transmitting radio simulators shall output Transmitter and Signal PDUs to represent their state.

(2) Receiving radio simulators shall reproduce the received signal to the fidelity level required by that particular receiver simulation. This shall be accomplished using the parameters provided in the Transmitter and Signal PDUs.

(3) The location of the radio described by the PDUs shall be determined either from the radio's Transmitter PDU or, if greater accuracy is required, from the Entity State PDU for the entity of which the radio is a part.

(4) A DIS Entity shall be associated with every radio transmitter and receiver. This entity shall be identified in each Transmitter or Receiver PDU. This entity may represent the radio itself or the vehicle that contains the radio.

(5) Receiving radio simulators may issue Receiver PDUs to reflect receiver state. These PDUs are for use by radio network monitors, data loggers and similar systems.

4.4.8.2 Transmitter PDU. The Transmitter PDU shall be used to communicate the state of a particular radio transmitter.

4.4.8.2.1 Information Contained in the Transmitter PDU. The Transmitter PDU shall contain the following information:

(1) Standard PDU header

(2) Identification of the entity that contains the radio transmitter

(3) Identification of the particular transmitter that is being described

(4) Identification of the type of transmitter that is being described

(5) State of the transmitter (whether it is off, on but not transmitting, or on and transmitting)

(6) Source of the radio input (whether the pilot, co-pilot, first officer, etc.)

(7) Location of the radiating portion of the antenna in both world and entity coordinates.

(8) Type of representation used for the radiation pattern from the antenna.

(9) Center frequency for transmission

(10) Bandwidth of the particular transmitter measured between the half-power (-3dB) points. This value represents total bandwidth, not the deviation from the center frequency.

(11) Average power being transmitted

(12) Type of modulation used for transmission. This includes the spread-spectrum usage, details on modulation type, and the compatibility of the emissions from the subject transmitting device.

(13) Specification of the crypto or secure voice equipment if utilized. Also provides information to ensure that the transmitting and receiving crypto gear are utilizing the same crypto key.

(14) Modulation type specific parameters which define the details of the RF modulation used.

(15) Antenna Pattern parameters that describe the radiation pattern from the antenna, its orientation in space, and the polarization of the radiation.

4.4.8.2.2 Issuance of the Transmitter PDU. A Transmitter PDU shall be issued by a radio simulator when:

(1) A predetermined default time has elapsed. Different default time intervals shall be used by stationary and moving transmitters. This default time shall be modifiable by the simulation manager.

(2) Any parameter in the Transmitter PDU other than Antenna Location, Antenna azimuth, or Antenna elevation has changed.

(3) The difference between the current antenna location and the Antenna Location in the last issued Transmitter PDU exceeds a predefined positional threshold.

(4) The difference between the current antenna direction (azimuth and elevation) and the antenna direction

reported in the last issued Transmitter PDU exceeds a predefined orientation threshold.

(5) Simulated time has been paused.

(6) Simulated time has been restarted after a pause.

When a transmission is initiated, a Transmitter PDU shall be issued before the first Signal PDU of the transmission. When a transmission is concluded, a Transmitter PDU shall follow the final Signal PDU of the transmission.

The Transmitter PDU shall be issued using a real-time, best effort, multicast communication service.

4.4.8.2.3 Receipt of the Transmitter PDU. Upon receipt of the Transmitter PDU, the receiving radio simulator shall determine the effects of the transmission on the receiving radio. The effects may include the clear reception of the modulated signal, the addition of noise or jamming effects to signals already being received, or other simulator specific effects. If the receiver parameters are such that the radio transmission is at least partially received and demodulated, the receiving radio simulator may use this as a cue to begin processing the Signal PDUs from the received transmitter.

The location of the radio transmitter's antenna in world coordinates may be determined by either of two methods. If the first method is selected, the location of the antenna is given by the "antenna location" field of the Transmitter PDU. As an optional extension to this method, the antenna location may be extrapolated based on the locations in previously received Transmitter PDUs. The algorithm used to perform this extrapolation must guarantee that the extrapolated location does not deviate from the location in the last Transmitter PDU by more than the threshold described in 4.4.8.2.2(3).

If the second method is selected, the location of the antenna is calculated using a combination of Transmitter PDU's "relative antenna location" field and the dead reckoned model of the entity's location as computed from the corresponding Entity State PDUs (see 4.4.2.1.2).

4.4.8.3 Signal PDU. The Signal PDU shall be used to convey the audio or digital data carried by the simulated radio transmission.

4.4.8.3.1 Information Contained in the Signal PDU. The Signal PDU contains the content of a radio transmission. This content may be digitized voice, binary data, or an index into a database which defines the signal.

The Signal PDU shall contain the following information:

(1) Standard PDU header

(2) Identification of the entity that is the source of the transmission

(3) Identification of the particular transmitter that is transmitting

(4) Specification of the encoding scheme utilized. The encoding scheme is comprised of an encoding class (voice, raw data, application-specific data or pointer) and an encoding type.

For large scale interoperability of voice communication, 8-bit mu-law encoding sampled at 8 kHz (defined in Recommendation G.711, CCITT Blue Book - Fasciscle III.4) shall be supported by all DIS radio simulators. In addition, other encoding schemes may be negotiated between a set of radio simulators. The mechanism for this negotiation is outside the scope of the RCP. These negotiated encoding schemes may be utilized to support the special requirements of particular radio simulation types, to reduce network loading, or to provide for higher fidelity.

In order to reduce bandwidth consumption in certain C3 applications, transmission of actual message content may be replaced with pointers. These pointers shall be interpreted by receivers as indices into a standardized database of pre-recorded frames of audio, text, etc. The details of the database are not addressed here.

(5) The sample rate in samples per second for voice digital data and the baud rate for data

(6) The length of the data fields expressed in bits

(7) The number of individual voice samples

4.4.8.3.2 Issuance of the Signal PDU. A Signal PDU shall be issued whenever voice or data is being transmitted. The first Signal PDU of a radio transmission shall be issued after the Transmitter PDU that indicates the beginning of the transmission. The final Signal PDU of a radio transmission shall precede the Transmitter PDU which indicates the end of the transmission. The encoding class may change within the Signal PDUs of a single transmission; However, the encoding type for a particular radio shall not change for the duration of a simulation exercise.

The Signal PDU shall be issued using a real-time, best effort, multicast communication service.

4.4.8.3.3 Receipt of the Signal PDU. Upon receipt of the Signal PDU, the receiving entity shall determine if its simulation is able to detect or interpret the transmission. If the transmission is detected, the receiving entity then simulates the receipt of the voice or data.

4.4.8.4 Receiver PDU. The Receiver PDU may be used to communicate the state of a particular radio receiver. Its primary application is in communicating state information to radio network monitors, data loggers, and similar applications for use in debugging, supervision and after-action review.

4.4.8.4.1 Information Contained in the Receiver PDU. The Receiver PDU shall contain the following information:

(1) Standard PDU header.

(2) Identification of the entity that is controlling the radio receiver.

(3) Identification of the particular receiver that is being described.

(4) State of the receiver (whether it is off, on but not receiving, or on and receiving).

(5) Identification of the entity that is controlling the radio transmitter.

(6) Identification of the particular transmitter that is being described.

(7) Average power being received.

4.4.8.4.2 Issuance of the Receiver PDU. A Receiver PDU, when applicable to a simulation, shall be issued by a radio simulator when:

(1) A predetermined default time has elapsed. Different default time intervals shall be used by stationary and moving transmitters. This default time shall be set by the simulation manager.

(2) Any parameter in the Receiver PDU has changed.

(3) Simulated time has been paused.

(4) Simulated time has been restarted after a pause.

The Receiver PDU shall be issued using a real-time, best effort, multicast communication service.

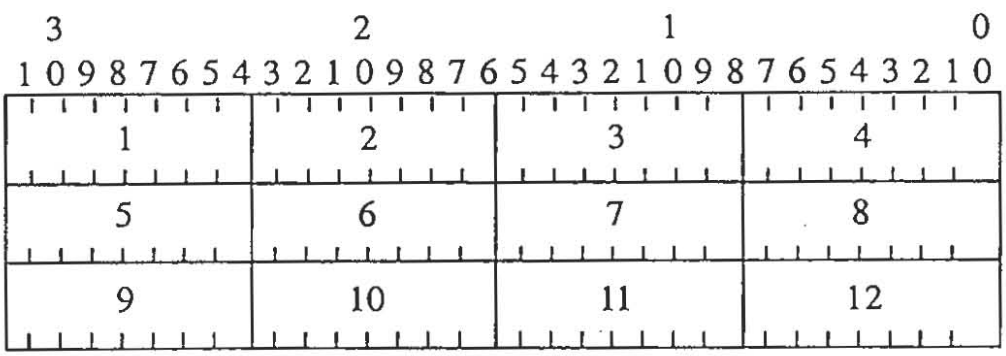
4.4.8.4.3 Receipt of the Receiver PDU. No positive response to a Receiver PDU shall be required of a radio simulation. Applications for which these PDU are useful may respond appropriately.

5. Detailed Requirements

5.1 Introduction. This section defines the PDUs and their fields.

5.2 Representation of Data. Octet ordering, enumeration representation and number representation are described in the following paragraphs.

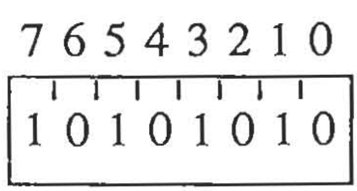
5.2.1 Octet Ordering. The order of transmission of the data described in this standard shall be resolved to the octet level. In Fig 5-1, the order of the transmission of octets is illustrated. The octets in Fig 5-1 shall be transmitted in the order in which they are numbered.



0243-1972

Figure 5-1
Transmission Order of Octets

Whenever an octet represents a numeric quantity, the left most bit in the diagram shall represent the high order or most significant bit. That is, the bit labeled 7 is the most significant bit. For example, Fig 5-2 represents the decimal value 170.



0243-1973

Figure 5-2
Significance of Bits

Similarly, whenever a multi-octet field represents a numeric quantity, the left most bit of the whole field shall be the most significant bit. When a multi-octet quantity is transmitted, the most significant octet is transmitted first.

5.2.2 Enumeration Representation. All enumerated types shall begin with zero for the first element of the enumeration. Enumerations may have any size between 1 and 32 bits.

5.2.3 Number Representation. Numbers shall be represented as either floating point numbers or integers.

5.2.3.1 Floating Point Numbers. Single and double precision floating point numbers shall adhere to the IEEE 754-1985 Standard.

5.2.3.2 Integers. Integers shall be represented as signed or unsigned. Signed integers shall be represented in two's-complement form, where the most significant bit shall designate the sign bit. This bit shall have a value of zero for positive numbers and one for negative numbers. Integers may have a size of 8, 16, or 32 bits.

5.3 Basic Data Types and Records. This section specifies requirements for basic data types and records. Enumeration values and bit-encoded values are given in Section 4 in Document IST-CR-93-02 unless otherwise stated.

5.3.1 Angle Representation. Angles shall be specified as 32-bit floating point numbers expressed in radians.

5.3.2 Angular Velocity Vector Record. The angular velocity of simulated entities shall be represented by the Angular Velocity Vector Record. This record shall specify the rate at which an entity's orientation is changing. This rate shall be measured in radians per second measured about each of the entity's own coordinate axes. The record shall consist of three fields. The first field shall represent velocity about the x-axis, the second about the y-axis, and the third about the z-axis (see 5.3.20.1). The positive direction of the angular velocity is defined by the right-hand rule. The Angular Velocity Vector Record is represented in Fig 5-3.

Rate about x-axis	32-bit floating point
Rate about y-axis	32-bit floating point
Rate about z-axis	32-bit floating point

Fig 5-3
Angular Velocity Vector Record

5.3.3 Articulation Parameter Record. The specification of articulation parameters for movable parts and attached parts of an entity shall be represented by an Articulation Parameter Record. This record shall specify whether or not a change has occurred, the Part ID of the articulated part to which it is attached, and the type and value of each parameter (see Annex A for further explanation, figures, and examples).

5.3.3.1 Change Indicator. The change of any parameter for any articulated part shall be indicated by a change indicator field. This field shall be specified by a 16-bit unsigned integer. This field shall be set to zero for each exercise and sequentially incremented by one for each change in articulation parameters. In the case where all possible values are exhausted, the numbers shall be reused beginning at zero.

5.3.3.2 ID - Part Attached to. The identification of the articulated part to which this articulation parameter is attached shall be specified by a 16-bit unsigned integer. This field shall contain the value zero if the articulated part is attached directly to the entity.

5.3.3.3 Parameter Type Record. The type of parameter represented shall be specified by a 32-bit enumeration. Parameter types are defined in Annex A.

5.3.3.4 Parameter Value. The parameter value shall be specified by a 64-bit field. The definition of the 64 bits shall be determined based on the type of parameter specified in the parameter type field (see 5.3.3.3).

The Articulation Parameters Record is represented in Fig 5-4.

Change Indicator	16-bit unsigned integer
ID - Part Attached to	16-bit unsigned integer
Parameter Type	32-bit Parameter Type Record
Parameter Value	64 bits

Fig 5-4
Articulation Parameter Record

5.3.4 Boolean. A boolean data type shall be represented as a single bit representing a true-false value. This bit shall represent an enumeration type of one bit, where the value 0 is interpreted as false, and the value 1 as true.

5.3.5 Burst Descriptor Record. The firing of a round or a burst of ammunition shall be represented by a Burst Descriptor Record. This record shall specify the type of munition fired, the type of warhead, the type of fuze, the number of rounds fired, and the rate at which the rounds are fired in rounds per second. The fields of this record are described in the subparagraphs that follow.

5.3.5.1 Munition. The munition type shall be specified by an Entity Type Record defined in 5.3.10.

5.3.5.2 Warhead. The warhead shall be specified by a 16-bit enumeration (see Section 1 in Document IST-CR-93-02).

5.3.5.3 Fuze. The fuze shall be specified by a 16-bit enumeration (see Section 1 in Document IST-CR-93-02).

5.3.5.4 Quantity and Rate. Quantity and rate each shall be specified by 16-bit unsigned integers. Quantity shall represent the number of rounds fired in the burst, and rate shall represent the rounds per minute for the munition specified. For quantity equal to one, the rate field shall contain zeros. The Burst Descriptor Record is represented in Fig 5-5.

Munition	Entity Type 64-bit Entity Type Record
Warhead	16-bit enumeration
Fuze	16-bit enumeration
Quantity	16-bit unsigned integer
Rate	16-bit unsigned integer

Fig 5-5
Burst Descriptor Record

5.3.6 Entity Appearance Record. The appearance of an entity shall be specified by an Entity Appearance Record. This record shall be defined as a 32-bit record of enumerations. The values defined for this record are included in Section 3 in Document IST-CR-93-02.

5.3.7 Entity Capabilities Record. The capabilities of an entity shall be specified by an Entity Capabilities Record. This record shall be defined as a 32-bit record of boolean types. The values defined for this record are included in Section 4 in Document IST-CR-93-02.

5.3.8 Entity Identifier Record. The unique identification of each entity in an exercise shall be specified by an Entity Identifier Record. This identification number shall consist of a Simulation Address Record (see 5.3.8.1) and an entity identification number. No entity shall be assigned an ID containing all zeros or all ones. The fields of this record are described in the subparagraphs that follow.

5.3.8.1 Simulation Address Record. An entity's simulation address shall be specified by a Simulation Address Record. A Simulation Address Record shall consist of the site identification number and the host identification number. These fields are described in 5.3.8.1.1 and 5.3.8.1.2. The Simulation Address Record is represented in Fig 5-6.

5.3.8.1.1 Site Identifier. Each DIS site shall be assigned a unique site identifier. No site shall be assigned an ID containing all zeros or all ones. The mechanism by which site IDs are assigned is outside the scope of this standard. This identifier shall be specified by a 16-bit unsigned integer.

5.3.8.1.2 Application Identifier. Each simulation application at a DIS site shall be assigned a application identifier unique within that site. No simulation application shall be assigned an ID containing all zeros or all ones. One or more simulation applications may be resided in a single host computer. The mechanism by which application IDs are assigned is outside the scope of this standard. This identifier shall be specified by a 16-bit unsigned integer.

Site Identifier	16-bit unsigned integer
Application Identifier	16-bit unsigned integer

Fig 5-6
Simulation Address Record

5.3.8.2 Entity Identifier. Each entity in a given exercise executing on a DIS application shall be assigned an entity identifier unique within that application. This identifier is valid for the duration of the exercise, however, entity IDs shall be reused when all possible entity IDs have been exhausted. No entity shall have an ID of zero. This number need not be registered or retained for future exercises. The Entity Identifier shall be represented using a 16-bit unsigned integer.

The Entity Identifier Record is represented in Figure 5-7.

Simulation Address	Site ID: 16-bit unsigned integer Application ID: 16-bit unsigned integer
Entity Identifier	16-bit unsigned integer

Fig 5-7
Entity Identifier Record

5.3.9 Entity Marking Record. Entity markings shall be specified by the Entity Marking Record. This record shall specify the character set used in the marking and the string of characters to be interpreted for display. The character set shall be specified by an 8-bit enumeration. The string of characters shall be represented by an 11 element character string. This string shall begin with the most significant octet located at the lowest address. Characters not used shall contain an integer value zero. The Entity Marking Record is represented in Fig 5-8.

Character Set	8-bit enumeration
1st Character	8-bit unsigned integer
2nd Character	8-bit unsigned integer
•	•
•	•
11th Character	8-bit unsigned integer

Fig 5-8
Entity Marking Record

5.3.10 Entity Type Record. The type of entity in a DIS exercise shall be specified by an Entity Type Record. This record shall specify the kind of entity, the country of design, the domain, the specific identification of the entity, and any extra information necessary for describing the entity. Fields not used shall contain the value zero. These fields are described below. The Entity Type Record is represented in Fig 5-9.

5.3.10.1 Kind. This field shall identify the kind of entity described by the Entity Type Record. This field shall be represented by an 8-bit enumeration. Values for this field are found in Section 6 in Document IST-CR-93-02.

5.3.10.2 Domain. This field shall specify the domain in which the entity operates (for example, subsurface, surface, land) except for munition entities. For munition entities this field shall specify the domain of the target (for example, the munition might be a surface-to-air, so the domain would be anti-air). This field shall be represented by an 8-bit enumeration. Values for this field are found in Section 6 in Document IST-CR-93-02.

5.3.10.3 Country. This field shall specify the country to which the design of the entity is attributed. This field shall be represented by a 16-bit enumeration. Values for this field are found in Section 4 in Document IST-CR-93-02.

5.3.10.4 Category. This field shall specify the main category that describes the entity. This field shall be represented by an 8-bit enumeration. Values for this field are found in Section 6 in Document IST-CR-93-02.

5.3.10.5 Subcategory. This field shall specify a particular subcategory to which an entity belongs based on the category field. This field shall be represented by an 8-bit enumeration. Values for this field are found in Section 6 in Document IST-CR-93-02.

5.3.10.6 Specific. This field shall specify specific information about an entity based on the subcategory field. This field shall be represented by an 8-bit enumeration. Values for this field are found in Section 6 in Document IST-CR-93-02.

5.3.10.7 Extra. This field shall specify extra information required to describe a particular entity. The contents of this field shall depend on the type of entity represented. This field shall be represented by an 8-bit enumeration.

Entity Kind	8-bit enumeration
Domain	8-bit enumeration
Country	16-bit enumeration
Category	8-bit enumeration
Subcategory	8-bit enumeration
Specific	8-bit enumeration
Extra	8-bit enumeration

Fig 5-9
Entity Type Record

5.3.11 Euler Angles Record. Orientation of a simulated entity shall be specified by the Euler Angles Record. This record shall specify three angles as described in 3.2(12) and Figure 1-3. These angles shall be specified with respect to the entity's coordinate system. The three angles shall each be specified by a 32-bit floating point number representing radians. The Euler Angles Record is represented in Fig 5-10.

PSI	32-bit floating point
THETA	32-bit floating point
PHI	32-bit floating point

Fig 5-10
Euler Angles Record

5.3.12 Event Identifier Record. Event identification shall be specified by the Event Identifier Record. This record shall consist of a Simulation Address Record and a 16-bit unsigned integer specifying the event number. The latter is uniquely assigned within the host by the simulation application that initiates the sequence of events. The Event Identifier Record shall be set to one for each exercise and incremented by one for each fire event or collision event. In the case where all possible values are exhausted, the numbers may be reused beginning again at one. The Event Identifier Record is represented in Fig 5-11.

Simulation Address	Site ID: 16-bit unsigned integer Application ID: 16-bit unsigned integer
Event Identifier	16-bit unsigned integer

Fig 5-11
Event Identifier Record

5.3.13 Exercise Identifier. Exercise identification shall be specified by an 8-bit unsigned integer value. This value shall be unique to each exercise occurring simultaneously on the same communications medium. (See Section 5.3.15[2]).

5.3.14 Force ID. This field shall distinguish the different teams or sides in a DIS exercise. This field shall be specified by an 8-bit enumeration. Values defined for this field are found in Section 4 in Document IST-CR-93-02.

5.3.15 Protocol Data Unit (PDU) Header Record. A PDU header record shall be the first part of each PDU. This record is represented in Fig 5-12. The fields of the PDU header record are described in the following four items (see also 4.4.1).

(1) Protocol version - This field shall specify the version of protocol used in this PDU. Protocol Data Units found in this standard document shall be specified as version 3. This field shall be specified by an 8-bit enumeration.
(2) Exercise identification - This field shall specify the exercise to which the PDU pertains. The value contained in this field shall not be equal to zero. This field shall be represented by an Exercise Identifier (see 5.3.13).
(3) Protocol Data Unit type - This field shall indicate the type of PDU that follows. This field shall be represented by an 8-bit enumeration. The values in this field are defined in Section 4 in document IST-CR-93-02.
(4) Time Stamp - This field shall specify the time which the data in the PDU is valid. This field shall be represented by a timestamp (see 5.3.19).
(5) Length - This field shall specify the length of the PDU in octets. This field shall be represented by an 16-bit unsigned integer.

PROTOCOL VERSION	8-bit enumeration
EXERCISE IDENTIFIER	8-bit unsigned integer
PDU TYPE	8-bit enumeration
PADDING	8 bits unused
TIME STAMP	32-bit unsigned integer
LENGTH	16-bit unsigned integer
PADDING	16 bits unused

Fig 5-12
Protocol Data Unit Header Record

5.3.16 Repair Type. Repair types shall be specified by a 16-bit enumeration. Values defined for this field are found in Section 2 in Document IST-CR-93-02.

5.3.17 Service Type. Service types shall be specified by an 8-bit enumeration. Values defined for this field are found in Section 4 in Document IST-CR-93-02.

5.3.18 Supply Quantity Record. Supply quantity shall be represented by the Supply Quantity Record. This record shall contain fields specifying the type of supply and the quantity of that supply. These fields are described in 5.3.18.1 and 5.3.18.2.

5.3.18.1 Supply Type. The supply type field shall be specified by an Entity Type Record (see 5.3.10 and Section 1 in Document IST-CR-93-02).

5.3.18.2 Quantity. The quantity field shall be specified by a 32-bit floating point number representing the number of units of a supply type. The unit measure depends on the supply type and shall use the SI units of measure used for such supplies. The Supply Quantity Record is represented in Fig 5-13.

Supply Type	Entity Type: 64-bit record
Quantity	32-bit floating point

Fig 5-13
Supply Quantity Record

5.3.19 Timestamp. Time stamping shall be used to indicate the time at which the data contained in the PDU is valid. This timestamp shall be specified using a 32-bit unsigned integer representing units of time passed since the beginning of the current hour. The least significant bit shall indicate whether the timestamp is absolute or relative.

5.3.19.1 Absolute Timestamp. An absolute timestamp shall be used when simulation application clocks are synchronized to Universal Coordinated Time (UTC). The use of the absolute timestamp shall be signified by the least significant bit set to one.

5.3.19.2 Relative Timestamp. A relative timestamp shall be used when simulation application clocks are not synchronized. Each simulation application shall keep time beginning with an arbitrary starting point. The time indicated by the timestamp shall be relative to the simulation application issuing the PDU. The use of the relative timestamp shall be signified by the least significant bit set to zero.

5.3.19.3 Scale. The scale of the time value contained in the most significant 31 bits of the timestamp shall be determined by setting one hour equal to $(2^{31}-1)$, thereby resulting in each time unit representing $3600\text{sec}/(2^{31}-1) = 1.676..$ microseconds.

5.3.20 Vector Record. Vector values for entity coordinates, linear acceleration, and linear velocity, shall be represented using a Vector Record. This record shall consist of three fields, each a 32-bit floating point number. The unit of measure represented by these fields shall depend on the information represented. The values utilizing the Vector Record are described in detail below. The Vector Record is represented in Fig 5-14.

5.3.20.1 Entity Coordinate Vector. Location with respect to a particular entity shall be specified with respect to three orthogonal axes whose origin shall be the center of the bounding volume of the entity excluding its articulated and attached parts (see Figure 1-2). The x-axis extends in the positive direction out the front of the entity. The y-axis extends in the positive direction out the right side of the entity as viewed from above, facing in the direction of the positive x-axis. The z-axis extends in the positive direction downward. Each vector component shall represent meters from the origin (see Figure 1-2).

5.3.20.2 Linear Acceleration Vector. Linear acceleration shall be represented as a vector with components in either world coordinate system or entity's coordinate system depending on the value in the Dead Reckoning Algorithm field. Each vector component shall represent acceleration in meters per second squared.

5.3.20.3 Linear Velocity Vector. Linear velocity shall be represented as a vector with components in either world coordinate system or entity's coordinate system depending on the value in the Dead Reckoning Algorithm field. Each vector component shall represent velocity in meters per second.

First Vector Component	32-bit floating point
Second Vector Component	32-bit floating point
Third Vector Component	32-bit floating point

Fig 5-14
Vector Record

5.3.21 World Coordinates Record. Location of the origin of the entity's coordinate system shall be specified by a set of three coordinates: X, Y, and Z. The shape of the earth shall be specified using WGS 84. The origin of this coordinate system shall be the centroid of the earth, with the X-axis passing through the Prime Meridian at the Equator, the Y-axis passing through 90 degrees East longitude at the Equator, and the Z-axis passing through the North pole (see Figure 1-1). These coordinates shall represent meters from the centroid of the earth. A 64-bit double precision floating point number shall represent the location for each coordinate. The World Coordinates Record is represented in Fig 5-15.

X - Coordinate	64-bit floating point
Y - Coordinate	64-bit floating point
Z - Coordinate	64-bit floating point

Fig 5-15
World Coordinates Record

5.4 Protocol Data Units for Distributed Interactive Simulation

5.4.1 Introduction. Section 5.4 lists and describes the Protocol Data Units required by this standard. Note that in the figures in Section 5.4, each field starts at an offset from the beginning of the PDU which is a multiple of its size. Each record starts at an offset which is a multiple of the size of the largest field in the record. The length of the PDU is a multiple of the size of the largest field in the PDU and 32 bits. Padding bits shall be used to achieve these offsets. All padding fields shall be zero.

5.4.2 List of DIS Protocol Data Units

(1) Entity Information
- (a) Entity State
  - (i) Entity State PDU

(2) Entity Interaction
- (a) Weapons Fire
  - (i) Fire PDU
  - (ii) Detonation PDU
- (b) Logistics Support
  - (i) Service Request PDU
  - (ii) Resupply Offer PDU
  - (iii) Resupply Received PDU
  - (iv) Resupply Cancel PDU
  - (v) Repair Complete PDU
  - (vi) Repair Response PDU
- (c) Collisions
  - (i) Collision PDU

(3) Simulation Management
- (a) Initialization and Termination
  - (i) Create Entity PDU
  - (ii) Remove Entity PDU
  - (iii) Start/Resume PDU
  - (iv) Stop/Freeze PDU
  - (v) Acknowledge PDU
- (b) Action Command
  - (i) Action Request PDU
  - (ii) Action Response PDU
- (c) Data Acquisition and Distribution
  - (i) Data Query PDU
  - (ii) Set Data PDU
  - (iii) Data PDU
  - (iv) Event Report PDU
  - (v) Message PDU

(4) Distributed Emission Regeneration
- (a) Emission
  - (i) Emission PDU
  - (ii) Laser PDU

(5) Radio Communication
- (a) Radio Transmitter and Signal
  - (i) Transmitter PDU

(ii) Signal PDU
(iii) Receiver PDU

5.4.3 Entity Information. (see also 4.4.2)

5.4.3.1 Entity State PDU. Information about a particular entity shall be communicated by issuing an Entity State PDU. The Entity State PDU shall contain the following fields:

(1) PDU Header - This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header Record (see 5.3.15).
(2) Entity Identification - This field shall identify the entity issuing the PDU. This field shall be represented by an Entity Identifier Record (see 5.3.8).
(3) Force Identification - This field shall identify the force to which the issuing entity belongs. This field shall be represented by an 8-bit enumeration (see Section 4 in Document IST-CR-93-02).
(4) Number of Articulation Parameters - This field shall specify the number of articulation parameters required for the specification of the presence and state of all articulated parts. This field shall be represented by an 8-bit unsigned integer (see Annex A).
(5) Entity Type - This field shall identify the entity type to be displayed by members of the same force as the issuing entity. This field shall be represented by an Entity Type Record (see 5.3.10, and Section 6 in Document IST-CR-93-02).
(6) Alternate Entity Type - This field shall identify the entity type to be displayed by members of forces other than that of the issuing entity. This field shall be represented by an Entity Type Record (see 5.3.10 and Section 4 in Document IST-CR-93-02).
(7) Entity Location - This field shall specify an entity 's physical location in the simulated world. This field shall be represented by a World Coordinates Record (see 5.3.21).
(8) Entity Linear Velocity - This field shall specify an entity's linear velocity. This field shall be represented by a Linear Velocity Vector Record (see 5.3.20.3).
(9) Entity Orientation - This field shall specify an entity's orientation. This field shall be represented by an Euler Angles Record (see 5.3.11).
(10) Entity Appearance - This field shall specify the dynamic changes to the entity's appearance attributes. This field shall be represented by an Entity Appearance Record. The values for the field contained in this record are defined in Section 3 in Document IST-CR-93-02.

(11) Dead Reckoning Parameters - This field will be used to provide parameters for dead reckoning the position and orientation of the entity. Dead Reckoning algorithm in use, Entity Acceleration, and Angular Velocity shall be included as part of the dead reckoning parameters. One-hundred twenty bits are reserved for other parameters that are currently undefined.
   (a) Dead Reckoning Algorithm - This field shall specify the dead reckoning algorithm in use by the issuing entity. This field shall be represented by an 8-bit enumeration. Values for this enumeration are defined in Section 7 in Document IST-CR-93-02.
   (b) Other Parameters - This field shall specify other required dead reckoning parameters to be determined. This field shall consist of 120 bits.
   (c) Entity Linear Acceleration - This field shall specify an entity's linear acceleration. This field shall be represented by a Linear Acceleration Vector Record (see 5.3.20.2).
   (d) Entity Angular Velocity - This field shall specify an entity's angular velocity. This field shall be represented by an Angular Velocity Vector Record (see 5.3.2).

(12) Entity Marking - This field shall identify any unique markings on an entity (for example, a bumper number or country symbol). This field shall be represented by an Entity Marking Record (see 5.3.9).

(13) Capabilities - This field shall specify the entity's capabilities. This field shall be represented by an Entity Capabilities Record (see 5.3.7 and Section 4 in Document IST-CR-93-02).

(14) Articulation Parameters - This field shall specify the parameter values for representation of each articulated or attached part. Each articulation shall be represented by an Articulation Parameter Record (see 5.3.3 and Annex A).

The Entity State PDU is represented in Fig 5-16.

FIELD SIZE (bits)	ENTITY STATE PDU FIELDS	
96	PDU HEADER	Protocol Version - 8-bit enumeration
		Exercise ID - 8-bit unsigned integer
		PDU Type - 8-bit enumeration
		Padding - 8 bits unused
		Time Stamp - 32-bit unsigned integer
		Length - 16-bit unsigned integer
		Padding - 16 bits unused
48	ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
8	FORCE ID	8-bit enumeration
8	# OF ARTICULATION PARAMETERS	8-bit unsigned integer
64	ENTITY TYPE	Entity Kind - 8-bit enumeration
		Domain - 8-bit enumeration
		Country - 16-bit enumeration
		Category - 8-bit enumeration
		Subcategory - 8-bit enumeration
		Specific - 8-bit enumeration
		Extra - 8-bit enumeration
64	ALTERNATIVE ENTITY TYPE	Entity Kind - 8-bit enumeration
		Domain - 8-bit enumeration
		Country - 16-bit enumeration
		Category - 8-bit enumeration
		Subcategory - 8-bit enumeration
		Specific - 8-bit enumeration
		Extra - 8-bit enumeration
192	ENTITY LOCATION	X - Component - 32-bit floating point
		Y - Component - 32-bit floating point
		Z - Component - 32-bit floating point
96	ENTITY LINEAR VELOCITY	X - Component - 64-bit floating point
		Y - Component - 64-bit floating point
		Z - Component - 64-bit floating point

8 bit Padding removed

Moved to front

Time stamp moved to the header

0049-1274.7

FIGURE 5-16
Entity State PDU.

FIELD SIZE (bits)	ENTITY STATE PDU FIELDS (CONT'D)	
96	ENTITY ORIENTATION	Psi - 32-bit floating point
		Theta - 32-bit floating point
		Phi - 32-bit floating point
32	ENTITY APPEARANCE	32-bit record of enumerations
320	DEAD RECKONING PARAMETERS	Dead Reckoning Algorithm - 8-bit enumeration
		Other Parameters - 120 bits unused
		Entity Linear Accel - 3x32-bit floating points
		Entity Angular Velocity - 3x32-bit integers
96	ENTITY MARKING	Character set - 8-bit enumeration
		11 - 8-bit unsigned integers
32	CAPABILITIES	32 Boolean fields
n x 128	ARTICULATION PARAMETERS	Change - 16-bit unsigned integer
		ID - attached to - 16-bit unsigned integer
		Parameter type - 32-bit parameter type record
		Parameter value - 64-bit

For each Articulation Parameter

fields order are swapped

24 bit Padding

0049-1274.8

Total Entity State PDU size = (1152 + 128n) bits
where n = number of articulation parameters

Fig 5-16 (Continued)
Entity State PDU

5.4.4 Entity Interaction. The following paragraphs describe entity interaction data, including weapon fire, logistics support and collision.

5.4.4.1 Weapons Fire (see also 4.4.3)

5.4.4.1.1 Fire PDU. The firing of a weapon shall be communicated by issuing a Fire PDU. The Fire PDU shall contain the following fields:

(1) PDU Header - This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header Record (see 5.3.15).
(2) Firing Entity Identification - This field shall identify the firing entity. This field shall be represented by an Entity Identifier Record (see 5.3.8).
(3) Target Entity Identification - This field shall identify the intended target. If the intended target is unknown, this field shall contain zeros. This field shall be represented by an Entity Identifier Record (see 5.3.8).
(4) Munition Identification - This field shall specify the entity ID of the fired munition if tracking data is required. A munition ID shall have a value of zero if tracking data for the munition is not required. This field shall be represented by an Entity Identifier Record (see 5.3.8).
(5) Event Identification - This field shall contain a number generated by the firing entity to associate related events. This field shall be represented by an Event Identifier Record (see 5.3.12).
(6) Location in World Coordinate - This field shall specify the location, in world coordinates, from which the munition was launched. This field shall be represented by a World Coordinates Record (see 5.3.21).
(7) Burst Descriptor - This field shall describe the type of munition fired, the warhead, the fuze, the quantity, and the rate. This field shall be represented by a Burst Descriptor Record (see 5.3.5).
(8) Velocity - This field shall specify speed and direction of the fired munition as it leaves its launcher. This field shall be represented by a Linear Velocity Vector Record (see 5.3.20.3).
(9) Range - This field shall specify the range that an entity's fire control system has assumed in computing the fire control solution. This field shall be represented by a 32-bit floating point number in meters. For systems where range is unknown or unavailable, this field shall contain a value of zero.

The Fire PDU is represented in Fig 5-17.

FIELD SIZE (bits)	FIRE PDU FIELDS	
96	PDU HEADER	Protocol Version - 8-bit enumeration
		Exercise ID - 8-bit unsigned integer
		PDU Type - 8-bit enumeration
		Padding - 8 bits unused
		Time Stamp - 32-bit unsigned integer
		Length - 16-bit unsigned integer
		Padding - 16 bits unused
48	FIRING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
48	TARGET ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
48	MUNITION ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
48	EVENT-ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Event - 16-bit unsigned integer
32	PADDING	32-bit unsigned integer
192	LOCATION IN WORLD	X-coordinate- 64-bit floating point
		Y-coordinate- 64-bit floating point
		Z-coordinate- 64-bit floating point
128	BURST DESCRIPTOR	Munition - 64-bit Entity Type Record
		Warhead - 16-bit enumeration
		Fuze - 16-bit enumeration
		Quantity - 16-bit unsigned integer
		Rate -16-bit unsigned integer
96	VELOCITY	X-component - 32-bit floating point
		Y-component - 32-bit floating point
		Z-component - 32-bit floating point
32	RANGE	32-bit floating point

Previously this was Time Stamp

0049-0960.2

Total Fire PDU Size = 768 bits

Fig 5-17
Fire PDU

5.4.4.1.2 Detonation PDU. The detonation or impact of munitions shall be communicated by issuing a Detonation PDU. The Detonation PDU shall contain the following fields:

(1) PDU Header - This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header Record (see 5.3.15).
(2) Firing Entity Identification - This field shall identify the firing entity. This field shall be represented by an Entity Identifier Record (see 5.3.8).
(3) Target Entity Identification - This field shall identify the target entity. If the target ID is unknown, this field shall contain the value zero. This field shall be represented by an Entity Identifier Record (see 5.3.8).
(4) Munition Identification - This field shall specify the entity ID of the fired munition if tracking data is required. A munition ID shall have a value of zero if tracking data for the munition is not required. This field shall be represented by an Entity Identifier Record (see 5.3.8).
(5) Event Identification - This field shall contain the same number as the event identifier of the Fire PDU that communicated the launch of the munition. If the detonation is not preceded by a corresponding fire event, then the event identifier field of the Event Identifier record shall be zero (e.g., land mines detonation). This field shall be represented by an Event Identifier Record (see 5.3.12).
(6) Velocity - This field shall specify the velocity of the munition immediately before detonation/impact. This field shall be represented by a Linear Velocity Vector Record (see 5.3.20.3).
(7) Location in World Coordinate - This field shall specify the location of the detonation in world coordinates. This field shall be represented by a World Coordinates Record (see 5.3.21).
(8) Burst Descriptor - This field shall describe the type of munition fired, the warhead, the fuze, the quantity, and the rate. This field shall be represented by a Burst Descriptor Record (see 5.3.5).
(9) Location in Entity's coordinates - This field shall specify the location of the detonation or impact in the target entity's coordinate system. This information should be used for damage assessment. This field shall be represented by an Entity Coordinate Vector Record (see 5.3.20.1). If the ID of the target is unknown, this field shall contain zeros.
(10) Detonation Result - This field shall specify the result of the detonation. This field shall be represented by an 8-bit enumeration (see Section 4 in Document IST-CR-93-02).

(11) Number of Articulation Parameters - This field shall specify the number of articulation parameters required for the specification of the presence and state of all relevant articulated or attached parts (see 4.4.3.3.4). This field shall be represented by an 8-bit unsigned integer (see Annex A).

(12) Articulation Parameters - This field shall specify the parameter values for representation of each relevant articulated or attached part (see 4.4.3.3.4). Each articulation shall be represented by an Articulation Parameter Record (see 5.3.3 and Annex A).

The Detonation PDU is represented in Fig 5-18.

FIELD SIZE (bits)	DETONATION PDU FIELDS	
96	PDU HEADER	Protocol Version - 8-bit enumeration
		Exercise ID - 8-bit unsigned integer
		PDU Type - 8-bit enumeration
		Padding - 8 bits unused
		Time Stamp - 32-bit unsigned integer
		Length - 16-bit unsigned integer
		Padding - 16 bits unused
48	FIRING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
48	TARGET ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
48	MUNITION ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
48	EVENT ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Event - 16-bit unsigned integer
96	VELOCITY	X - component - 32-bit floating point
		Y - component - 32-bit floating point
		Z - component - 32-bit floating point
192	LOCATION IN WORLD	X-coordinate - 64-bit floating point
		Y-coordinate - 64-bit floating point
		Z-coordinate - 64-bit floating point
128	BURST DESCRIPTOR	Muntion - 64-bit Entity Type Record
		Warhead - 16-bit enumeration
		Fuze - 16-bit enumeration
		Quantity - 16-bit unsigned integer
		Rate - 16-bit unsigned integer

time stamp in
the header

locity moved
the front.

0049-0961.2

Fig 5-18
Detonation PDU

FIELD SIZE (bits)	DETONATION PDU FIELDS (CONT'D)	
96	LOCATION IN ENTITY COORDINATES	x - coordinate - 32-bit floating point
		y - coordinate - 32-bit floating point
		z - coordinate - 32-bit floating point
8	DETONATION RESULT	8-bit enumeration
8	# OF ARTICULATION PARAMETERS	8-bit unsigned integer
16	PADDING	16 bits unused
n x 128	ARTICULATION PARAMETERS	Change - 16-bit unsigned integer
		ID - attached to - 16-bit unsigned integer
		Parameter type - 32-bit Parameter Type Record
		Parameter value - 64 bits

For each Articulation Parameter

0049-0961.3

Total Detonation PDU Size = (832 + 128n) bits
where n = number of articulation parameters

Fig 5-18 (Continued)
Detonation PDU

5.4.4.2 Logistics Support (see also 4.4.4)

5.4.4.2.1 Service Request PDU. A request for logistics support shall be communicated by issuing a Service Request PDU. The Service Request PDU shall consist of the following fields:

(1) PDU Header - This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header Record (see 5.3.15).
(2) Requesting Entity Identification - This field shall identify the entity that is requesting the service. This field shall be represented by an Entity Identifier Record (see 5.3.8).
(3) Servicing Entity Identification - This field shall identify the entity that is able to provide the service requested. This field shall be represented by an Entity Identifier Record (see 5.3.8).
(4) Service Type Requested - This field shall describe the type of service being requested. This field shall be represented by a Service Type (see Section 4 in Document IST-CR-93-02).
(5) Number of Supply Types - For a service of resupply, this field shall specify the number of different supplies being requested. If the service requested is not resupply, this field shall contain the value zero. This field shall be represented by an 8-bit unsigned integer.
(6) Supplies - For a service of resupply, this field shall specify the type of supply and the amount of that supply for the number of supplies specified above. If the service requested is not resupply, this field shall not be present. This field shall be represented by a Supply Quantity Record (see 5.3.18).

The Service Request PDU is represented in Fig 5-19.

FIELD SIZE (bits)	SERVICE REQUEST PDU FIELDS	
96	PDU HEADER	Protocol Version - 8-bit enumeration
		Exercise ID - 8-bit unsigned integer
		PDU Type - 8-bit enumeration
		Padding - 8 bits unused
		Time Stamp - 32-bit unsigned integer
		Length - 16-bit unsigned integer
		Padding - 16 bits unused
48	REQUESTING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
48	SERVICING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
8	SERVICE TYPE REQUESTED	8-bit enumeration
8	NUMBER OF SUPPLY TYPES	8-bit unsigned integer
16	PADDING	16 bits unused
n x 96	SUPPLIES	Entity Kind - 8-bit enumeration
		Domain - 8-bit enumeration
		Country - 16-bit enumeration
		Category - 8-bit enumeration
		Subcategory - 8-bit enumeration
		Specific - 8-bit enumeration
		Extra - 8-bit enumeration
		Quantity - 32-bit floating point

For each Supply Type

0049-0964.1

Total Service Request PDU size = (224 + 96n) bits
where n = number of supply types

Fig 5-19
Service Request PDU

5.4.4.2.2 Resupply Offer PDU. The offering of supplies shall be communicated by issuing a Resupply Offer PDU. The Resupply Offer PDU shall contain the following fields:

(1) PDU Header - This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header Record (see 5.3.15).
(2) Receiving Entity Identification - This field shall identify the receiving entity. This field shall be represented by an Entity Identifier Record (see 5.3.8).
(3) Supplying Entity Identification - This field shall identify the supplying entity. This field shall be represented by an Entity Identifier Record (see 5.3.8).
(4) Number of Supply Types - This field shall specify the number of different supply types being offered. This field shall be represented by an 8-bit unsigned integer.
(5) Supplies - This field shall specify the type of supply and the amount of that supply for each of the supply types specified above. This field shall not be present if the Number of Supply Types field value is zero. This field shall be represented by a Supply Quantity Record (see 5.3.18).

The Resupply Offer PDU is represented in Fig 5-20.

FIELD SIZE (bits)	RESUPPLY OFFER PDU FIELDS	
96	PDU HEADER	Protocol Version - 8-bit enumeration
		Exercise ID - 8-bit unsigned integer
		PDU Type - 8-bit enumeration
		Padding - 8 bits unused
		Time Stamp - 32-bit unsigned integer
		Length - 16-bit unsigned integer
		Padding - 16 bits unused
48	RECEIVING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
48	SUPPLYING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
8	NUMBER OF SUPPLY TYPES	8-bit unsigned integer
24	PADDING	24 bits unused
n x 96	SUPPLIES	Entity Kind - 8-bit enumeration
		Domain - 8-bit enumeration
		Country - 16-bit enumeration
		Category - 8-bit enumeration
		Subcategory - 8-bit enumeration
		Specific - 8-bit enumeration
		Extra - 8-bit enumeration
		Quantity - 32-bit floating point

For each Supply Type

0049-0967.1

Total Resupply Offer PDU size = (224 + 96n) bits
where n = number of supply types

Fig 5-20
Resupply Offer PDU

5.4.4.2.3 Resupply Received PDU. The receipt of supplies shall be communicated by issuing a Resupply Received PDU. The Resupply Received PDU shall contain the following fields:

(1) PDU Header - This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header Record (see 5.3.15).
(2) Receiving Entity Identification - This field shall identify the receiving entity. This field shall be represented by an Entity Identifier Record (see 5.3.8).
(3) Supplying Entity Identification - This field shall identify the supplying entity. This field shall be represented by an Entity Identifier Record (see 5.3.8).
(4) Number of Supply Types - This field shall specify the number of different supplies taken by the receiving entity. This field shall be represented by an 8-bit unsigned integer.
(5) Supplies - This field shall specify the type of supply and the amount of that supply for each of the supply types specified above. This field shall not be present if the Number of Supply Types field value is zero. This field shall be represented by a Supply Quantity Record (see 5.3.18).

The Resupply Received PDU is represented in Fig 5-21.

FIELD SIZE (bits)	RESUPPLY RECEIVED PDU FIELDS	
96	PDU HEADER	Protocol Version - 8-bit enumeration
		Exercise ID - 8-bit unsigned integer
		PDU Type - 8-bit enumeration
		Padding - 8 bits unused
		Time Stamp - 32-bit unsigned integer
		Length - 16-bit unsigned integer
		Padding - 16 bits unused
48	RECEIVING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
48	SUPPLYING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
8	NUMBER OF SUPPLY TYPES	8-bit unsigned integer
24	PADDING	24 bits unused
n x 96	SUPPLIES	Entity Kind - 8-bit enumeration
		Domain - 8-bit enumeration
		Country - 16-bit enumeration
		Category - 8-bit enumeration
		Subcategory - 8-bit enumeration
		Specific - 8-bit enumeration
		Extra - 8-bit enumeration
		Quantity - 32-bit floating point

For each Supply Type

0049-0966.1

Total Resupply Received PDU size = (224 + 96n) bits
where n = number of supply types

Figure 5-21
Resupply Received PDU

5.4.4.2.4 Resupply Cancel PDU. The canceling of a service function by either the receiving or the supplying entity shall be communicated by issuing a Resupply Cancel PDU. The Resupply Cancel PDU shall contain the following fields:

(1) PDU Header - This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header Record (see 5.3.15).
(2) Receiving Entity Identification - This field shall identify the entity that has requested the resupply service. This field shall be represented by an Entity Identifier Record (see 5.3.8).
(3) Supplying Entity Identification - This field shall identify the supplying entity. This field shall be represented by an Entity Identifier Record (see 5.3.8).

The Resupply Cancel PDU is represented in Fig 5-22.

FIELD SIZE (bits)	RESUPPLY CANCEL PDU FIELDS	
96	PDU HEADER	Protocol Version - 8-bit enumeration
		Exercise ID - 8-bit unsigned integer
		PDU Type - 8-bit enumeration
		Padding - 8 bits unused
		Time Stamp - 32-bit unsigned integer
		Length - 16-bit unsigned integer
		Padding - 16 bits unused
48	RECEIVING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
48	SUPPLYING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer

0049-0965.1

Total Resupply Cancel PDU size = 192 bits

Fig 5-22
Resupply Cancel PDU

5.4.4.2.5 Repair Complete PDU. When a Service Request PDU has been received and the repairing entity has completed a requested repair, the repairing entity shall notify the receiving entity of the repair by issuing a Repair Complete PDU. The Repair Complete PDU shall contain the following fields:

(1) PDU Header - This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header Record (see 5.3.15).
(2) Receiving Entity Identification - This field shall identify the entity that is requesting repairs. This field shall be represented by an Entity Identifier Record (see 5.3.8).
(3) Repairing Entity Identification - This field shall identify the repairing entity. This field shall be represented by an Entity Identifier Record (see 5.3.8).
(4) Repair - This field shall describe the repair performed. If the repairing entity is unable to provide or complete the needed repair, this field shall have a value of zero.

If the repairing entity makes all of the needed repairs, this field shall have a value of one. This field shall be represented by a repair type (see 5.3.16 and Section 2 in Document IST-CR-93-02).

The Repair Complete PDU is represented in Fig 5-23.

FIELD SIZE (bits)	REPAIR COMPLETE PDU FIELDS	
96	PDU HEADER	Protocol Version - 8-bit enumeration
		Exercise ID - 8-bit unsigned integer
		PDU Type - 8-bit enumeration
		Padding - 8 bits unused
		Time Stamp - 32-bit unsigned integer
		Length - 16-bit unsigned integer
		Padding - 16 bits unused
48	RECEIVING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
48	REPAIRING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
16	REPAIR	16-bit enumeration
16	PADDING	16 bits unused

0049-0969.1

Total Repair Complete PDU size = 224 bits

FIG 5-23
Repair Complete PDU

5.4.4.2.6 Repair Response PDU. When a receiving entity receives a Repair Complete PDU from its repairing entity, the receiving entity shall acknowledge the receipt of the repair by issuing a Repair Response PDU. The Repair Response PDU shall contain the following fields:

(1) PDU Header - This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header Record (see 5.3.15).
(2) Receiving Entity Identification - This field shall identify the entity that requested repairs. This field shall be represented by an Entity Identifier Record (see 5.3.8).
(3) Repairing Entity Identification - This field shall identify the repairing entity. This field shall be represented by an Entity Identifier Record (see 5.3.8).
(4) Repair Result - This field shall specify the result of the repair specified in the Repair Complete PDU. This field shall be represented by an 8-bit enumeration (see Section 4 in Document IST-CR-93-02).

The Repair Response PDU is represented in Fig 5-24.

FIELD SIZE (bits)	REPAIR RESPONSE PDU FIELDS	
96	PDU HEADER	Protocol Version - 8-bit enumeration
		Exercise ID - 8-bit unsigned integer
		PDU Type - 8-bit enumeration
		Padding - 8 bits unused
		Time Stamp - 32-bit unsigned integer
		Length - 16-bit unsigned integer
		Padding - 16 bits unused
48	RECEIVING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
48	REPAIRING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
8	REPAIR RESULT	8-bit enumeration
24	PADDING	24 bits unused

0049-1438

Total Repair Response PDU size = 224 bits

Fig 5-24
Repair Response PDU

5.4.4.3 Collisions. (see also 4.4.5)

5.4.4.3.1 Collision PDU. Collisions between entities shall be communicated by issuing a Collision PDU. The Collision PDU shall contain the following fields:

(1) PDU Header - This field shall contain data common t all DIS PDUs. The PDU Header shall be represented by the PDU Header Record (see 5.3.15).
(2) Issuing Entity Identification - This field shall identify the entity that is issuing the PDU. This field shall be represented by an Entity Identifier Record (see 5.3.8).
(3) Colliding Entity Identification - This field shall identify the entity which has collided with the issuing entity. If the entity ID is unknown or the collision is with a terrain object, this field shall contain zeros. This field shall be represented by an Entity Identifier Record (see 5.3.8).
(4) Event Identification - This field shall contain a number generated by the issuing simulation application to associate related events. This field shall be represented by an Event Identifier Record (see 5.3.12).
(5) Velocity - This field shall contain the velocity (at the time the collision is detected) of the issuing entity. This field shall be represented by the Linear Velocity Vector Record (see 5.3.20.3).
(6) Mass - This field shall contain the mass of the issuing entity. This field shall be represented by a 32-bit floating point number representing kilograms.
(7) Location - This field shall specify the location of the collision with respect to the entity with which the issuing entity collided. This field shall be represented by an Entity Coordinate Vector Record (see 5.3.20.1).

The Collision PDU is represented in Fig 5-25.

FIELD SIZE (bits)	COLLISION PDU FIELDS	
96	PDU HEADER	Protocol Version - 8-bit enumeration
		Exercise ID - 8-bit unsigned integer
		PDU Type - 8-bit enumeration
		Padding - 8 bits unused
		Time Stamp - 32-bit unsigned integer
		Length - 16-bits unsigned integer
		Padding - 16 bits unused
48	ISSUING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
48	COLLIDING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
48	EVENT ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Event - 16-bit unsigned integer
16	PADDING	16 bits unused
96	VELOCITY	X-Component - 32-bit floating point
		Y-Component - 32-bit floating point
		Z-Component - 32-bit floating point
32	MASS	32-bit floating point
96	LOCATION (with respect to Entity)	x-Component - 32-bit floating point
		y-Component - 32-bit floating point
		z-Component - 32-bit floating point

0049-0970.1

Total Collision PDU size = 480 bits

Fig 5-25
Collision PDU

5.4.5 Simulation Management. (see 4.4.6)

5.4.5.1 Basic Data Types and Records for Simulation Management. This section specifies requirements for basic data types and records to support simulation management. Enumeration values and bit-encoded values are given in Section 4 of document IST-CR-93-2 unless otherwise stated.

5.4.5.1.1 Simulation Management PDU Header Record. A Simulation Management PDU header Record shall be the first part of each simulation management PDU. The PDU header described in 5.3.15 is part of this header record. The Simulation Management PDU header record contains information required for all simulation management PDUs. This Simulation Management PDU header Record is represented in Figure 5-26. The fields of the PDU Header Record are described below.

(1) PDU Header - This field contains data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header Record described in 5.3.15.

(2) Originating Entity Identification - This field shall identify the entity issuing the Start PDU. This field shall be represented by an Entity Identifier Record (see 5.3.12).

(3) Receiving Entity Identification - This field shall identify the entity to which the Start PDU is intended. This field shall be represented by an Entity Identifier Record (see 5.3.12).

PDU Header	Protocol Version - 8-bit unsigned integer Exercise ID - 8-bit unsigned integer PDU Type - 8-bit enumeration Padding - 8 bits unused Time Stamp - 32-bit unsigned integer Length - 16-bit unsigned integer Padding - 8 bits unused
Originating Entity ID	Site - 16-bit unsigned integer Application - 16-bit unsigned integer Entity - 16-bit unsigned integer Group - 16-bit unsigned integer
Receiving Entity ID	Site - 16-bit unsigned integer Application - 16-bit unsigned integer Entity - 16-bit unsigned integer Group - 16-bit unsigned integer

Fig 5-26
Simulation Management PDU Header Record

5.4.5.1.2 Fixed Datum Record. Fixed Datum information shall be represented using the Fixed Datum Record. This record shall specify the fixed datum type and the value for that fixed datum type. The fields of this record are described in the subparagraphs that follow.

5.4.5.1.2.1 Fixed Datum ID. The type of fixed datum to be communicated shall be specified with a Fixed Datum ID field. This field shall consist of a 32-bit enumeration.

5.4.5.1.2.2 Fixed Datum Value. The value for a particular Fixed Datum ID shall be specified with a Fixed Datum value field. This field shall consist of a 32-bit field. The field type shall depend on the type of fixed datum as specified by the Fixed Datum ID. The padding for possible unused bits shall be appended to the Fixed Datum Value.

The Fixed Datum Record is represented in Figure 5-27.

Fixed Datum ID	32-bit enumeration
Fixed Datum Value	32 bits (varies)

Fig 5-27
Fixed Datum Record

5.4.5.1.3 Variable Datum Record. Variable Datum information shall be represented using the Variable Datum Record. This record shall specify the variable datum type, the datum length and the value for that variable datum type. The fields of this record are described in the subparagraphs that follow.

5.4.5.1.3.1 Variable Datum ID. The type of variable datum to be communicated shall be specified with a Variable Datum ID field. This field shall consist of a 32-bit enumeration.

5.4.5.1.3.2 Variable Datum Value. The value for a particular Variable Datum ID shall be specified with a Variable Datum Value field. This field shall consist of a multiple of 64-bit field. The field type shall depend on the type of datum as specified by the Variable Datum ID. The padding for possible unused bits shall be appended to the Variable Datum Value.

The Variable Datum Record is represented on Figure 5-28.

Variable Datum ID	32-bit enumeration
Variable Datum Length	32 bits integer
Variable Datum Value	Multiple of 64-bits (varies)

Fig 5-28
Variable Datum Record

5.4.5.1.4 Datum Specification Record. Datum information shall be represented using the Datum Specification Record. This record shall specify the number of Fixed Datum records and Variable Datum record, the length of the variable datum fields, and the actual values. The fields of this record are described in the subparagraphs that follow.

(1) Number of Fixed Datum Field - This field shall specify the number of fixed datum fields. This field shall be represented by a 32-bit unsigned integer.

(2) Number of Variable Datum Field - This field shall specify the number of variable datum fields required to supply database names or character fields or bit streams which exceed 64 bits. This fields shall be represented by a 32-bit unsigned integer.

(3) Fixed Datum Values - These fields shall specify the types of fixed datum and their value. These fields shall be represented by Fixed Datum Records.

(4) Variable Datum Values - These fields shall specify the types of variable datum, their length and their value. This field shall be represented by Variable Datum Records.

The Datum Specification Record is represented in Figure 5-29.

Number of Fixed Datum Records	32-bit unsigned integer
Number of Variable Datum Records	32-bit unsigned integer
Fixed Datum Record #1	Datum ID - 32-bit enum Datum Value - 32 bits
Fixed Datum Record #2	Datum ID - 32-bit enum Datum Value - 32 bits
. . .	
Fixed Datum Record #N	Datum ID - 32-bit enum Datum Value - 32 bits
Variable Datum Record #1	Variable Datum ID - 32-bit enum Variable Datum Length - 32-bit integer Variable Datum Value - multiple of 64 bits
Variable Datum Record #2	Variable Datum ID - 32-bit enum Variable Datum Length - 32-bit integer Variable Datum Value - multiple of 64 bits
. . .	
Variable Datum Record #M	Variable Datum ID - 32-bit enum Variable Datum Length - 32-bit integer Variable Datum Value - multiple of 64 bits

Fig 5-29
Datum Specification Record

5.4.5.1.5 Clock Time Record. Time measurements that surpasses one hour shall be represented by a Clock Time Record. The fields of this record are described in the subparagraphs that follow.

(1) Hours - This field shall specify the hours since January 1, 1970. This field shall be represented by a 32-bit integer.

(2) Time Past the Hours - This field shall specify the time past the hour indicated in Hours field. This field shall be represented by a timestamp (see 5.3.19)

The Clock Time Record is represented in Figure 5-30.

Hour	32-bit integer
Time Past the Hour	32-bit unsigned integer

Fig 5-30
Clock Time Record

5.4.5.2 Create Entity PDU. The creation of a new entity shall be communicated using a Create Entity PDU. This PDU shall consist of a Simulation Management PDU Header:

Simulation Management PDU Header - These fields shall identify the PDU header information, the originating entity, and the intended receiving entity. The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header Record described in 5.4.5.1.1.

The Create Entity PDU is represented in Figure 5-31.

FIELD SIZE (bits)	CREATE ENTITY PDU FIELDS	
96	PDU HEADER	Protocol Version - 8-bit enumeration
		Exercise ID - 8-bit unsigned integer
		PDU Type - 8-bit enumeration
		Padding - 8 bits unused
		Time Stamp - 32-bit unsigned integer
		Length - 16-bit unsigned integer
		Padding - 16 bits unused
64	ORIGINATING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
		Group - 16-bit unsigned integer
64	RECEIVING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
		Group - 16-bit unsigned integer

0049-2767

Total Create Entity PDU Size = 224 bits

Figure 5-31
Create Entity PDU

5.4.5.3 Remove Entity PDU. The removal of an entity from an exercise shall be communicated with a Remove Entity PDU. The Remove Entity PDU shall consist a Simulation Management PDU Header:

Simulation Management PDU Header - These fields shall identify the PDU header information, the originating entity, and the intended receiving entity. The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header Record described in 5.4.5.1.1.

The Remove Entity PDU is represented in Figure 5-32.

FIELD SIZE (bits)	REMOVE ENTITY PDU FIELDS	
96	PDU HEADER	Protocol Version - 8-bit enumeration
		Exercise ID - 8-bit unsigned integer
		PDU Type - 8-bit enumeration
		Padding - 8 bits unused
		Time Stamp - 32-bit unsigned integer
		Length - 16-bit unsigned integer
		Padding - 16 bits unused
64	ORIGINATING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
		Group - 16-bit unsigned integer
64	RECEIVING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
		Group - 16-bit unsigned integer

0049-2768

Total Remove Entity PDU Size = 224 bits

Figure 5-32
Remove Entity PDU

5.4.5.4 Start/Resume PDU. The Start/Resume of an exercise shall be communicated using a Start/Resume PDU. The Start/Resume PDU shall contain the following fields:

(1) Simulation Management PDU Header - These fields shall identify the PDU header information, the originating entity, and the intended receiving entity. The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header Record described in 5.4.5.1.1.

(2) Real-World Time - This field shall specify the real world time (with respect to Greenwich time) at which the entity is to start/resume the exercise. This information shall be used by the participating simulation applications to start/resume an exercise synchronously. This field shall be represented by a Clock Time Record (see 5.4.5.1.5).

(3) Simulation Time - This field shall specify the simulation time (time of day in the simulated world) at which the entity will start/resume the exercise. This field shall be represented by a Clock Time Record (see 5.4.5.1.5).

The Start/Resume PDU is represented in Figure 5-33.

FIELD SIZE (bits)	START/RESUME PDU FIELDS	
96	PDU HEADER	Protocol Version - 8-bit enumeration
		Exercise ID - 8-bit unsigned integer
		PDU Type - 8-bit enumeration
		Padding - 8 bits unused
		Time Stamp - 32-bit unsigned integer
		Length - 16-bit unsigned integer
		Padding - 16 bits unused
64	ORIGINATING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
		Group - 16-bit unsigned integer
64	RECEIVING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
		Group - 16-bit unsigned integer
64	REAL-WORLD TIME	Hour - 32-bit integer
		Time Past the Hour 32-bit unsigned integer
64	SIMULATION TIME	Hour - 32-bit integer
		Time Past the Hour 32-bit unsigned integer

0049-2769

Total Start/Resume PDU Size = 352 bits.

Figure 5-33
Start/Resume PDU

5.4.5.5 Stop/Freeze PDU. The stopping or freezing of an exercise shall be communicated using a Stop/Freeze PDU. The Stop/Freeze PDU shall contain the following fields:

(1) Simulation Management PDU Header - These fields shall identify the PDU header information, the originating entity, and the intended receiving entity. The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header Record described in 5.4.5.1.1.

(2) Real-World Time - This field shall specify the real world time (with respect to Greenwich time) at which the entity is to stop/freeze the exercise. This field shall be represented by a Clock Time Record (see 5.4.5.1.5).

(3) Reason - This field shall specify the reason that an entity or exercise was stopped/frozen. This field shall be represented by an 8-bit enumeration.

The Stop/Freeze PDU is represented in Figure 5-34.

FIELD SIZE (bits)	STOP/FREEZE PDU FIELDS	
96	PDU HEADER	Protocol Version - 8-bit enumeration
		Exercise ID - 8-bit unsigned integer
		PDU Type - 8-bit enumeration
		Padding - 8 bits unused
		Time Stamp - 32-bit unsigned integer
		Length - 16-bit unsigned integer
		Padding - 16 bits unused
64	ORIGINATING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
		Group - 16-bit unsigned integer
64	RECEIVING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
		Group - 16-bit unsigned integer
64	REAL-WORLD TIME	Hour - 32-bit integer
		Time Past the Hour 32-bit unsigned integer
8	REASON	8-bit enumeration
24	PADDING	24 bits unused

0049-7770.1

Total Stop/Freeze PDU Size = 320 bits.

Figure 5-34
Stop/Freeze PDU

5.4.5.6 Acknowledge PDU. The acknowledgement of the receipt of a Start/Resume PDU, Stop/Freeze PDU, Create Entity PDU, or a Remove Entity PDU shall be communicated by issuing an Acknowledge PDU. The Acknowledge PDU shall contain the following fields:

(1) Simulation Management PDU Header - These fields shall identify the PDU header information, the originating entity, and the intended receiving entity. The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header Record described in 5.4.5.1.1.

(2) Acknowledge Flag - This field shall indicate what type of message has been acknowledged. This field shall be represented by a 16-bit enumeration.

(3) Response Flag - This field shall indicate whether or not the receiving entity was able to comply with the request, and for what reason. This field shall be represented by a 16-bit enumeration.

The Acknowledge PDU is represented in Figure 5-35.

FIELD SIZE (bits)	ACKNOWLEDGE PDU FIELDS	
96	PDU HEADER	Protocol Version - 8-bit enumeration
		Exercise ID - 8-bit unsigned integer
		PDU Type - 8-bit enumeration
		Padding - 8 bits unused
		Time Stamp - 32-bit unsigned integer
		Length - 16-bit unsigned integer
		Padding - 16 bits unused
64	ORIGINATING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
		Group - 16-bit unsigned integer
64	RECEIVING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
		Group - 16-bit unsigned integer
16	ACKNOWLEDGE FLAG	16-bit enumeration
16	RESPONSE FLAG	16 bit enumeration

0049-3771.1

Total Acknowledge PDU Size = 256 bits.

Figure 5-35
Acknowledge PDU

5.4.5.7 Action Request PDU. A request from an SM to a managed entity to perform a specified action shall be communicated using an Action Request PDU. The Action Request PDU shall consist of the following fields:

(1) Simulation Management PDU Header - These fields shall identify the PDU header information, the originating entity, and the intended receiving entity. The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header Record described in 5.4.5.1.1.

(2) Request ID - This field shall identify the request being made by the simulation manager. This field shall be represented by a 32-bit unsigned integer.

(3) Action ID - This field shall specify the particular action that is requested by the simulation manager. This field shall be represented by a 32-bit enumeration.

(4) Datum Information - This field shall specify the types of datum and the value of the datum to be communicated. This field shall be represented by a Datum Specification Record (see 5.4.5.1.4).

The Action Request PDU is represented in Figure 5-36.

FIELD SIZE (bits)	ACTION REQUEST PDU FIELDS	
96	PDU HEADER	Protocol Version - 8-bit enumeration
		Exercise ID - 8-bit unsigned integer
		PDU Type - 8-bit enumeration
		Padding - 8 bits unused
		Time Stamp - 32-bit unsigned integer
		Length - 16-bit unsigned integer
		Padding - 16 bits unused
64	ORIGINATING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
		Group - 16-bit unsigned integer
64	RECEIVING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
		Group - 16-bit unsigned integer
32	PADDING	32 bits unused
32	REQUEST ID	32-bit unsigned integer
32	ACTION ID	32-bit unsigned integer
32	NUMBER OF FIXED DATUM FIELDS (N)	32-bit unsigned integer
32	NUMBER OF VARIABLE DATUM FIELDS (M)	32-bit unsigned integer
64	FIXED DATUM #1	Datum ID - 32-bit enumeration
		Datum Value - 32 unsigned integer
⋮		
64	FIXED DATUM #N	Datum ID - 32-bit enumeration
		Datum Value - 32 bits
$64 + K_1 + P_1$	VARIABLE DATUM #1	Datum ID - 32-bit enumeration
		Datum Length - 32 bit unsigned integer (K)
		Datum Value - 64 bit
		⋮
⋮		
$64 + K_M + P_M$	VARIABLE DATUM #M	Datum ID - 32-bit enumeration
		Datum Length - 32 bit unsigned integer (K)
		Datum Value - 64 bit
		⋮

0049-2775.1

$$\text{Size} = (384 + 64\,N + \sum_{i=1}^{M} (64 + K_i + P_i))\ \text{bits}$$

N = number of fixed datum fields
M = number of variable datum fields
K_i = length of variable datum value i in bits
P_i = number of unused bits in the last 64 bit segment of variable datum value i

Figure 5-36
Action Request PDU

5.4.5.8 Action Response PDU. When an entity receives an Action Request PDU, that entity shall acknowledge the receipt of the Action Request PDU with an Action Response PDU. The Action Response PDU shall contain the following fields:

(1) Simulation Management PDU Header - These fields shall identify the PDU header information, the originating entity, and the intended receiving entity. The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header Record described in 5.4.5.1.1.

(2) Request ID - This field shall identify this matching response to a request made by the simulation manager. This field shall be represented by a 32-bit unsigned integer.

(3) Request Status - This field shall identify the status of the requested action. This field shall be represented by a 32-bit enumeration.

(4) Datum Information - This field shall specify the types of datum and the value of the datum to be communicated. This field shall be represented by a Datum Specification Record (see 5.4.5.1.4).

The Action Response PDU is represented in Figure 5-37.

FIELD SIZE (bits)	ACTION RESPONSE PDU FIELDS	
96	PDU HEADER	Protocol Version - 8-bit enumeration
		Exercise ID - 8-bit unsigned integer
		PDU Type - 8-bit enumeration
		Padding - 8 bits unused
		Time Stamp - 32-bit unsigned integer
		Length - 16-bit unsigned integer
		Padding - 16 bits unused
64	ORIGINATING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
		Group - 16-bit unsigned integer
64	RECEIVING ENTITY ID	Site - 16-bit unsigned integer
		Application- 16-bit unsigned integer
		Entity - 16-bit unsigned integer
		Group - 16-bit unsigned integer
32	PADDING	32 bits unused
32	REQUEST ID	32-bit unsigned integer
32	REQUEST STATUS	32-bit enumeration
32	NUMBER OF FIXED DATUM FIELDS (N)	32-bit unsigned integer
32	NUMBER OF VARIABLE DATUM FIELDS (M)	32-bit unsigned integer
64	FIXED DATUM #1	Datum ID - 32-bit enumeration
		Datum Value - 32 unsigned integer
	⋮	
64	FIXED DATUM #N	Datum ID - 32-bit enumeration
		Datum Value - 32 bits
$64 + K_1 + P_1$	VARIABLE DATUM #1	Datum ID - 32-bit enumeration
		Datum Length - 32 bit unsigned integer (K)
		Datum Value - 64 bit
		⋮
	⋮	
$64 + K_M + P_M$	VARIABLE DATUM #M	Datum ID - 32-bit enumeration
		Datum Length - 32 bit unsigned integer (K)
		Datum Value - 64 bit
		⋮

0049-2776

$$\text{Size} = (384 + 64\,N + \sum_{i=1}^{M} (64 + K_i + P_i\,))\ \text{bits}$$

N = number of fixed datum fields
M = number of variable datum fields
K_i = length of variable datum value i in bits
P_i = number of unused bits in the last 64 bit segment of variable datum value i

Figure 5-37
Action Response PDU

5.4.5.9 Data Query PDU. A request for data from an entity shall be communicated by issuing a Data Query PDU. The Data Query PDU shall contain the following fields:

(1) Simulation Management PDU Header - These fields shall identify the PDU header information, the originating entity, and the intended receiving entity. The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header Record described in 5.4.5.1.1.

(2) Time Interval - This field shall specify the time interval between issues of Data PDUs. A value of zero in this field shall mean that the requested data should be sent once and not at any previously specified time interval. This field shall be represented by a timestamp (see 5.3.19).

(3) Request ID - This field shall identify the data query request being made by the Simulation Manager. This field shall be represented by a 32-bit unsigned integer.

(4) Datum Information - This field shall specify the types of datum and their value to be communicated. This field shall be represented by a Datum Specification Record (see 5.4.5.1.4).

The Data Query PDU is represented in Figure 5-38.

FIELD SIZE (bits)	DATA QUERY PDU FIELDS	
96	PDU HEADER	Protocol Version - 8-bit enumeration
		Exercise ID - 8-bit unsigned integer
		PDU Type - 8-bit enumeration
		Padding - 8 bits unused
		Time Stamp - 32-bit unsigned integer
		Length - 16-bit unsigned integer
		Padding - 16 bits unused
64	ORIGINATING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
		Group - 16-bit unsigned integer
64	RECEIVING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
		Group - 16-bit unsigned integer
32	REQUEST ID	32-bit unsigned integer
32	TIME INTERVAL	32-bit unsigned integer
32	NUMBER OF FIXED DATUM FIELDS (N)	32-bit unsigned integer
32	NUMBER OF VARIABLE DATUM FIELDS (M)	32-bit unsigned integer
32	FIXED DATUM ID #1	32-bit enumeration
	⋮	
32	FIXED DATUM ID #N	32-bit enumeration
32	VARIABLE DATUM ID #1	32-bit enumeration
	⋮	
32	VARIABLE DATUM ID #M	32-bit enumeration

0049-2772

Total Data Query PDU Size = (352 + 32N + 32M) bits.

N = number of fixed datum fields
M = number of variable datum fields

Figure 5-38
Data Query PDU

5.4.5.10 Set Data PDU. Initializing or changing internal state information shall be communicated using a Set Data PDU. The Set Data PDU shall consist of the following fields:

(1) Simulation Management PDU Header - These fields shall identify the PDU header information, the originating entity, and the intended receiving entity. The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header Record described in 5.4.5.1.1.

(2) Request ID - This field shall identify the set data request being made by the simulation manager. This field shall be represented by a 32-bit unsigned integer.

(3) Datum Information - This field shall specify the types of datum and their value to be communicated. This field shall be represented by a Datum Specification Record (see 5.4.5.1.4).

The Set Data PDU is represented in Figure 5-39.

FIELD SIZE (bits)	SET DATA PDU FIELDS	
96	PDU HEADER	Protocol Version - 8-bit enumeration
		Exercise ID - 8-bit unsigned integer
		PDU Type - 8-bit enumeration
		Padding - 8 bits unused
		Time Stamp - 32-bit unsigned integer
		Length - 16-bit unsigned integer
		Padding - 16 bits unused
64	ORIGINATING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
		Group - 16-bit unsigned integer
64	RECEIVING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
		Group - 16-bit unsigned integer
32	REQUEST ID	32-bit unsigned integer
32	NUMBER OF FIXED DATUM FIELDS (N)	32-bit unsigned integer
32	NUMBER OF VARIABLE DATUM FIELDS (M)	32-bit unsigned integer
64	FIXED DATUM #1	Datum ID - 32-bit enumeration
		Datum Value - 32 unsigned integer
⋮	⋮	⋮
64	FIXED DATUM #N	Datum ID - 32-bit enumeration
		Datum Value - 32 bits
$64 + K_1 + P_1$	VARIABLE DATUM #1	Datum ID - 32-bit enumeration
		Datum Length - 32 bit unsigned integer (K)
		Datum Value - 64 bit
		⋮
⋮	⋮	⋮
$64 + K_M + P_M$	VARIABLE DATUM #M	Datum ID - 32-bit enumeration
		Datum Length - 32 bit unsigned integer (K)
		Datum Value - 64 bit
		⋮

0049-2780

$$\text{Set Data PDU Size} = (320 + 64N + \sum_{i=1}^{M} (64 + K_i + P_i))\ \text{bits}$$

N = number of fixed datum fields
M = number of variable datum fields
K_i = length of variable datum value i in bits
P_i = number of unused bits in the last 64 bit segment of variable datum value i

Figure 5-39
Set Data PDU

5.4.5.11 Data PDU. Information issued in response to a Data Query PDU or Set Data PDU shall be communicated using a Data PDU. The Data PDU shall contain the following fields:

(1) Simulation Management PDU Header - These fields shall identify the PDU header information, the originating entity, and the intended receiving entity. The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header Record described in 5.4.5.1.1.

(2) Request ID - This field shall identify the matching response to a Data Query PDU or Data Set PDU made by the Simulation Manager. This field shall be represented by a 32-bit unsigned integer.

(3) Datum Information - This field shall specify the types of datum and their value to be communicated. This field shall be represented by a Datum Specification Record (see 5.4.5.1.4).

The Data PDU is represented in Figure 5-40.

FIELD SIZE (bits)	DATA PDU FIELDS	
96	PDU HEADER	Protocol Version - 8-bit enumeration
		Exercise ID - 8-bit unsigned integer
		PDU Type - 8-bit enumeration
		Padding - 8 bits unused
		Time Stamp - 32-bit unsigned integer
		Length - 16-bit unsigned integer
		Padding - 16 bits unused
64	ORIGINATING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
		Group - 16-bit unsigned integer
64	RECEIVING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
		Group - 16-bit unsigned integer
32	REQUEST ID	32-bit unsigned integer
32	NUMBER OF FIXED DATUM FIELDS (N)	32-bit unsigned integer
32	NUMBER OF VARIABLE DATUM FIELDS (M)	32-bit unsigned integer
64	FIXED DATUM #1	Datum ID - 32-bit enumeration
		Datum Value - 32 unsigned integer
	⋮	
64	FIXED DATUM #N	Datum ID - 32-bit enumeration
		Datum Value - 32 bits
$64 + K_1 + P_1$	VARIABLE DATUM #1	Datum ID - 32-bit enumeration
		Datum Length - 32 bit unsigned integer (K)
		Datum Value - 64 bit
		⋮
	⋮	
$64 + K_M + P_M$	VARIABLE DATUM #M	Datum ID - 32-bit enumeration
		Datum Length - 32 bit unsigned integer (K)
		Datum Value - 64 bit
		⋮

0049-2773

$$\text{Total Data PDU Size} = (320 + 64\,N + \sum_{i=1}^{M} (64 + K_i + P_i))\ \text{bits}$$

N = number of fixed datum fields
M = number of variable datum fields
K_i = length of variable datum value i in bits
P_i = number of unused bits in the last 64 bit segment of variable datum value i

Figure 5-40
Data PDU

5.4.5.12 Event Report PDU. A managed entity shall report the occurrence of a significant event to the simulation manager using an Event Report PDU. The Event Report PDU shall consist of the following fields:

(1) Simulation Management PDU Header - These fields shall identify the PDU header information, the originating entity, and the intended receiving entity. The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header Record described in 5.4.5.1.1.

(2) Event Type - This field shall specify the type of event that caused the issue of an Event PDU. This field shall be represented by a 32-bit enumeration.

(3) Datum Information - This field shall specify the types of datum and their value be communicated. This field shall be represented by a Datum Specification Record (see 5.4.5.1.4).

The Event Report PDU is represented in Figure 5-41.

FIELD SIZE (bits)	EVENT REPORT PDU FIELDS	
96	PDU HEADER	Protocol Version - 8-bit enumeration
		Exercise ID - 8-bit unsigned integer
		PDU Type - 8-bit enumeration
		Padding - 8 bits unused
		Time Stamp - 32-bit unsigned integer
		Length - 16-bit unsigned integer
		Padding - 16 bits unused
64	ORIGINATING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
		Group - 16-bit unsigned integer
64	RECEIVING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
		Group - 16-bit unsigned integer
32	EVENT TYPE	32-bit enumeration
32	NUMBER OF FIXED DATUM FIELDS (N)	32-bit unsigned integer
32	NUMBER OF VARIABLE DATUM FIELDS (M)	32-bit unsigned integer
64	FIXED DATUM #1	Datum ID - 32-bit enumeration
		Datum Value - 32 unsigned integer
	⋮	
64	FIXED DATUM #N	Datum ID - 32-bit enumeration
		Datum Value - 32 bits
$64 + K_1 + P_1$	VARIABLE DATUM #1	Datum ID - 32-bit enumeration
		Datum Length - 32 bit unsigned integer (K)
		Datum Value - 64 bit
		⋮
	⋮	
$64 + K_M + P_M$	VARIABLE DATUM #M	Datum ID - 32-bit enumeration
		Datum Length - 32 bit unsigned integer (K)
		Datum Value - 64 bit
		⋮

0049-2781

$$\text{Event Report PDU Size} = (320 + 64\,N + \sum_{i=1}^{M} (64 + K_i + P_i)) \text{ bits}$$

N = number of fixed datum fields
M = number of variable datum fields
K_i = length of variable datum value i in bits
P_i = number of unused bits in the last 64 bit segment of variable datum value i

Figure 5-41
Event Report PDU

5.4.5.13 Message PDU. Arbitrary messages (character strings) shall be entered into the data stream by using a Message PDU. The Message PDU shall consist of the following fields:

(1) Simulation Management PDU Header - These fields shall identify the PDU header information, the originating entity, and the intended receiving entity. The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header Record described in 5.4.5.1.1.

(2) Number of Variable Datum Fields - This field shall specify the number of variable datum fields required to supply database names or character fields which exceed 32 bits. This field shall be represented by a 32-bit unsigned integer.

(3) Variable Datum Values - These fields shall specify the types of Variable Datum, their length and their value. This field shall be represented by a Variable Datum Record (see 5.6.3.1.3).

The Message PDU is represented in Figure 5-42.

FIELD SIZE (bits)	MESSAGE PDU FIELDS	
96	PDU HEADER	Protocol Version - 8-bit enumeration
		Exercise ID - 8-bit unsigned integer
		PDU Type - 8-bit enumeration
		Padding - 8 bits unused
		Time Stamp - 32-bit unsigned integer
		Length - 16-bit unsigned integer
		Padding - 16 bits unused
64	ORIGINATING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
		Group - 16-bit unsigned integer
64	RECEIVING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
		Group - 16-bit unsigned integer
32	NUMBER OF VARIABLE DATUM FIELDS (M)	32-bit unsigned integer
$64 + K_1 + P_1$	VARIABLE DATUM #1	Datum ID - 32-bit enumeration
		Datum Length - 32 bit unsigned integer (K)
		Datum Value - 64 bit
	⋮	
$64 + K_M + P_M$	VARIABLE DATUM #M	⋮
		Datum ID - 32-bit enumeration
		Datum Length - 32 bit unsigned integer (K)
		Datum Value - 64 bit
		⋮

0049-2774

$$\text{Total Message PDU Size} = \left(256 + \sum_{i=1}^{M} (64 + K_i + P_i)\right) \text{ bits}$$

N = number of fixed datum fields
M = number of variable datum fields
K_i = length of variable datum value i in bits
P_i = number of unused bits in the last 64 bit segment of variable datum value i

Figure 5-42
Message PDU

5.4.6 Distributed Emission Regeneration. (See 4.4.7)

5.4.6.1 Basic Data Types for Distributed Emission Regeneration. This section specifies requirements for additional basic data types and records to support Distributed Emission Regeneration (DER). Enumeration values and bit-encoded values are given in Section 4 of document IST-CR-93-2 unless otherwise stated.

5.4.6.1.1 Emitter System Record. Information about a particular emitter shall be represented using an Emitter System Record. This record shall consist of four fields: Emitter Name, Function and Emitter ID Number. These fields are described below:

(1) Emitter Name - This field shall specify the emitter name for a particular emitter. This field shall be represented by a 16-bit enumeration.

(2) Function - This field shall specify the function for a particular emitter. Typical functions include Surface-to-Air Missile (SAM), ground surveillance radar, etc. This field is intended to help receiving entities determine if the Emission PDU is of interest to the systems simulated by that entity. This field shall be represented by an 8-bit enumeration.

(3) Emitter Identification Number - This field shall specify the emitter identification number relative to a specific system. This field shall be represented by an 8-bit unsigned integer. This field allows the differentiation of multiple systems on an entity even if in some instances two or more of the systems may be identical emitter types.

The Emitter System Record is represented on Figure 5-43.

Emitter Name	16-bit Enumeration
Function	8-bit Enumeration
Emitter ID Number	8-bit Unsigned Integer

Figure 5-43 Emitter System Record

5.4.6.1.2 Fundamental Parameter Data Record. The Fundamental Parameter Data Record contains regeneration parameters that are variable throughout a scenario dependent on the actions of the participants in the simulation. This field also provides basic parametric data that may be used to support low-fidelity simulations which may not have the processing capability to model a high-fidelity regeneration of emission beams. This record shall consist of ten fields. These fields are described below:

(1) Frequency - This field shall specify the center frequency of the emission in hertz. Frequency modulation for a particular emitter and mode shall be derived from database parametrics stored in the receiving entity. This field shall be represented by a 32-bit floating point number.

(2) Frequency Range - This field shall specify the bandwidth of the frequencies corresponding to the Frequency field. Thus, if, for operational purposes, the Frequency is supposed to be a single number, then the Frequency Range shall be zero. If the Frequency Range reflects a finite number, then it may indicate a range of agile RFs (unintentional or intentional) for a radar and it may indicate a jamming package 3 db bandwidth for a jammer. This field shall be represented by a 32-bit floating point number.

(3) Effective Radiated Power (ERP) - This field shall specify the ERP for the emission in dbm. For a radar or a noise jammer, this field shall indicate the peak of the transmitted power. Thus, it includes peak transmitter power, transmission line losses, and peak of the antenna gain. This field shall be represented by a 32-bit floating point number.

(4) Pulse Repetition Frequency (PRF) - This field specifies the average PRF of the emission in hertz. PRF modulation for a particular emitter and mode shall be derived from database and parameters stored in the receiving entity. This field shall be represented by a 32-bit floating point number.

(5) Pulse Width - This field shall specify the average pulse width of the emission in micro-seconds. Pulse modulation for a particular emitter and mode shall be derived form database parameters store in the receiving entity. This field shall be represented by a 32-bit floating point number.

(6) Beam Azimuth Center - This field in conjunction with the following three fields specify the beam azimuth and elevation centers and the corresponding half-angles required to describe the scan volume covered by the emitter beam scan. For systems in which the center beam can slew greater than TBD degrees per second or systems with rapidly relocated beams (i.e., phase array), this scan coverage shall represent the current field-of-regard of the system. For systems in which

the center beam can be slewed at TBD degrees per second or less, this scan coverage shall represent the current field-of-view of the beam. All angles shall be measured in relation to the emission coordinate system. The azimuth center for 360 degree scan systems shall be 0 degrees. It should be noted that the scan volume described does not take into account masking by the entity. Masking determinations are a part of the regeneration process and require that appropriate database information be processed by the receiving entity. This field and the following three fields shall be represented by 32-bit floating point numbers representing units of Radians.

(7) Beam Azimuth Sweep - see Beam Azimuth Center

(8) Beam Elevation Center - see Beam Azimuth Center

(9) Beam Elevation Sweep - see Beam Azimuth Center

(10) Beam Sweep Sync - This field is provided to allow a receiver to synchronize its regenerated scan pattern to that of the emitter. This field when employed shall specify the percentage of time a scan is through its pattern from its origin. The pattern and origin data shall be derived from database parameters. This field shall be represented by a 32-bit floating point number.

The Fundamental parameter Data Record is shown in Figure 5-44.

Frequency	32-bit Floating Point
Frequency Range	32-bit Floating Point
ERP	32-bit Floating Point
PRF	32-bit Floating Point
Pulse Width	32-bit Floating Point
Beam Azimuth Center	32-bit Floating Point
Beam Azimuth Sweep	32-bit Floating Point
Beam Elevation Center	32-bit Floating Point
Beam Elevation Sweep	32-bit Floating Point
Beam Sweep Sync	32-bit Floating Point

Figure 5-44. Fundamental Parameter Data Record

5.4.6.2 Emission PDU. Information about active electronic warfare (EW) emissions, acoustic emissions and active countermeasures shall be communicated using an Emission PDU. The Emission PDU shall contain the following fields:

(1) PDU Header - These fields shall identify the protocol version number, the exercise identifier, and the protocol data unit type. The PDU Header shall be represented by the PDU Header described in 5.3.15.

(2) Emitting Entity ID - This field shall identify the entity that is the source of the emission. This field shall be represented by an Entity Identifier Record (see 5.3.8.2).

(3) Event Identification - This field shall contain a number generated by the issuing simulator to associate related events. This field shall be represented by an Event Identifier Record (see 5.3.12).

(4) State Update Indicator - This field shall be used to indicate if the data in the PDU represents a state update or just data that has changed since issuance of the last Emitter PDU (relative to the identified entity and emission system(s)). This field shall be represented by an 8-bit enumeration.

(5) Number of Systems - This field shall specify the number of emission systems being described in the current PDU. One, several or all of the emission systems on a particular

entity may be described in a single Emission PDU[3]. This field shall be represented by an 8-bit unsigned integer. The following information shall be provided for each system:

(a) System Data Length - This field shall specify the length of this systems data in 32-bit words. This field shall be represented by an 8-bit unsigned integer.

(b) Number of Beams - This field shall specify the number of beams being described in the current PDU for the system being described. This field shall be represented by an 8-bit unsigned integer.

(c) Emitter System - This field shall specify information about a particular emitter's system. This field shall be represented by an Emitter System Record (see 5.4.6.1.1).

(d) Location - This field shall specify the location of the antenna beam source with respect the emitting entity's coordinate system. This location shall be the origin of the emission coordinate system which shall be parallel to the entity coordinate system. This field shall be represented by an Entity Coordinate Vector Record (see 5.3.20.1).

The following information shall be provided for each beam:

(i) Beam Data Length - This field shall specify the length of this beams data (including track/jam information) in 32-bit words. This field shall be represented by an 8-bit unsigned integer.

(ii) Beam ID Number - This field shall specify a unique emission database number assigned to differentiate between otherwise similar or identical emitter beams within an emitter system. This field shall be represented by an 8-bit unsigned integer.

(iii) Beam Parameter Index - This field shall specify a beam parameter index number that shall be used by receiving entities in

---

[3] It should be noted that caution must be exercised to ensure that the number of systems being described does not cause the PDU length to exceed the maximum PDU length of 255 thirty-two-bit words.

conjunction with the emitter name field to provide a pointer to the stored database parameters required to regenerate the beam. why should This field shall be represented by a 16-bit unsigned integer.

(iv) Fundamental Parameter Data - This field shall specify dynamic parameters of the emitter. This field shall be represented by a Fundamental parameter Data Record (see 5.4.6.1.2).

(v) Beam Function - This field shall specify the function of a particular beam. This field shall be represented by an 8-bit enumeration.

(vi) Number of Targets in Track/Jam Field - This field, in conjunction with the following field, provides a mechanism for an entity with an emitting system to identify targets that are being illuminated by a track beam or target emitters it is attempting to jam. When these fields are used in conjunction with a tracking emitter, the target identified may or may not be tracked by the emitting system. When employed by a tracking emitter, this field shall specify the number of targets that are being identified as in the track beam being described. When employed by a jamming emitter, this field shall specify the number of emitters the system is attempting to jam. This field shall be represented by an 8-bit unsigned integer.

(vii) High Density Track/Jam - This field shall be used to indicate whether or not the receiving entities can assume that all targets, in the scan pattern which the sending emission can track (for a phased array system) or jam (for a jamming system), are being tracked or jammed respectively. This field shall be represented by an 8-bit enumeration.

(viii) Jamming Mode Sequence - This field shall be used to identify one or multiple jamming techniques being applied. This field shall be represented by a 32-bit unsigned integer.

The following information shall be provided for each target:

- Track/Jam Field - This field shall identify the targets in a emitter track beam or emitters a system is attempting to jam. This field shall be represented by an Entity Identifier Record (see 5.2.8), an Emitter Identifier and a Beam Identifier. The Emitter and Beam Identifiers shall be specified by 8-bit unsigned integers. For a tracking emission, the Emitter and Beam Identifiers shall contain all zeros. For a jamming emission, these identifiers are defined as follows:

  - Entity Identifier - The entity identifier shall specify the target entity ID.

  - Emitter Identifier - The Emitter Identifier shall be the Emitter ID number of the emitter for which the jamming emission is intended.

  - Beam Identifier - The Beam Identifier shall be Beam ID number of the emitter beam for which the jamming emission is intended.

The Emission PDU is represented in Figure 5-45.

FIELD SIZE (bits)	EMISSION PDU FIELDS		
96	PDU HEADER	Protocol Version - 8-bit enumeration	
		Exercise ID - 8-bit unsigned integer	
		PDU Type - 8-bit enumeration	
		Padding - 8 bits unused	
		Time Stamp - 32-bit unsigned integer	
		Length - 16-bit unsigned integer	
		Padding - 16 bits unused	
48	EMITTING ENTITY ID	Site - 16-bit unsigned integer	
		Application - 16-bit unsigned integer	
		Entity - 16-bit unsigned integer	
48	EVENT ID	Site - 16-bit unsigned integer	
		Application - 16-bit unsigned integer	
		Entity - 16-bit unsigned integer	
8	STATE UPDATE INDICATOR	8-bit enumeration	
8	NUMBER OF SYSTEMS (N)	8-bit unsigned integer	
16	PADDING	16 bits unused	
VARIES *	SYSTEM DATA LENGTH	8-bit unsigned integer	160 Bits
	NUMBER OF BEAMS (M)	8-bit unsigned integer	
	PADDING	16 bits unused	
	EMITTER SYSTEM	Emitter Name - 16-bit enumeration	
		Function - 8-bit unsigned integer	
		Emitter ID Number - 8-bit unsigned integer	
	LOCATION (With respect to entity)	x - 32-bit floating point	
		y - 32-bit floating point	
		z - 32-bit floating point	
	BEAM DATA LENGTH	8-bit unsigned integer	416 Bits
	BEAM ID NUMBER	8-bit unsigned integer	
	BEAM PARAMETER INDEX	16-bit unsigned integer	
	FUNDAMENTAL PARAMETER DATA	Frequency - 32-bit floating point	
		Frequency Range - 32-bit floating point	
		ERP - 32-bit floating point	
		PRF - 32-bit floating point	
		Pulse Width - 32-bit floating point	
		Beam AZ Center - 32-bit floating point	
		Beam AZ Sweep - 32-bit floating point	
		Beam EL Center - 32-bit floating point	
		Beam EL Sweep - 32-bit floating point	
		Beam Sweep SYNC - 32-bit floating point	
	BEAM FUNCTION	8-bit unsigned integer	
	NUMBER OF TARGET IN TRACK/JAM FIELD (P)	8-bit unsigned integer	
	HIGH DENSITY TRACK/JAM	8-bit unsigned integer	
	PADDING	8-bit unsigned integer	
	JAMMING MODE SEQUENCE	32-bit unsigned integer	
	TRACK/JAM	Site - 16-bit unsigned integer	64 Bits
		Application - 16-bit unsigned integer	
		Entity - 16-bit unsigned integer	
		Emitter ID - 8-bit unsigned integer	
		Beam ID - 8-bit unsigned integer	

8 bits is Emission category

0049-2718.1

$$* \text{ Emission PDU Size} = 224 + 160N + \sum_{i=1}^{N} 416M_i + \sum_{j=1}^{Mi} 64P_j \text{ bits}$$

Where
N = number of systems
M_i = number of beams in system i
P_j = Number of targets in track/jam field in beam j

Figure 5-45
Emission PDU

This is an unapproved Standards Draft, subject to change.

5.4.6.3 Laser PDU. Lasing operations shall be communicated by issuing a Laser PDU. The Laser PDU shall contain the following fields:

(1) PDU Header - These fields shall identify the protocol version number, the exercise identifier, and the protocol data unit type. The PDU Header shall be represented by the PDU Header described in 5.3.15.

(2) Lasing Entity ID - This field shall identify the entity that is positioning the laser. This field shall be represented by an Entity Identifier Record (see 5.3.8.2).

(3) Code Name - This field shall identify the code name for the laser system. This field shall be represented by a 16-bit enumeration.

(4) Lased Entity ID - This field shall identify the entity that is currently being lased. This field shall contain zeros if the laser spot is not on an entity. This field shall be represented by an Entity Identifier Record (see 5.3.8.2).

(5) Laser Code - This field shall identify the laser code being used by the lasing entity. This field shall be represented by an 8-bit enumeration.

(6) Laser Power - This field shall identify the laser output power in watts. This field shall be represented by a 32-bit floating point number.

(7) Laser Wavelength - This field shall identify the laser wavelength in units of microns. This field shall be represented by a 32-bit floating point number.

(8) Laser Spot with respect to Lased Entity - This field shall specify the location of the laser spot with respect to the lased entity's coordinate system when the spot is on an entity. This field shall be represented by an Entity Coordinate Vector Record (see 5.3.20.1).

(9) Laser Spot Location - This field shall identify the location of the laser spot with respect to the world coordinate system. This field shall be represented by a World Coordinates Record (see 5.3.21).

The Laser PDU is represented in Figure 5-46.

FIELD SIZE (bits)	LASER PDU FIELDS	
96	PDU HEADER	Protocol Version - 8-bit enumeration
		Exercise ID - 8-bit unsigned integer
		PDU Type - 8-bit enumeration
		Padding - 8 bits unused
		Time Stamp - 32-bit unsigned integer
		Length - 16-bit unsigned integer
		Padding - 16 bits unused
48	LASING ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
16	CODE NAME	16 - bit enumeration
48	LASED ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
8	PADDING	8 bit unused
8	LASER CODE	8 - bit enumeration
32	LASER POWER	32-bit floating point
32	LASER WAVELENGTH	32-bit floating point
96	LASER SPOT WITH RESPECT TO LASED ENTITY	x-Coordinate - 32-bit floating point
		y-Coordinate - 32-bit floating point
		z-Coordinate - 32-bit floating point
192	LASER SPOT LOCATION	X-Coordinate - 64-bit floating point
		Y-Coordinate - 64-bit floating point
		Z-Coordinate - 64-bit floating point

0049-2765

Total Laser PDU Size = 576 bits

Figure 5-46
Laser PDU

5.4.6.4 Expendables PDU - TBD

5.4.6.5 IFF PDU - TBD

5.4.7 Radio Communications. (see 4.4.8)

5.4.7.1 Basic Data Types for Radio Communications. This section specifies requirements for additional basic data types and records to support Radio Communications. Enumeration values and bit-encoded values are given in Section 4 of document IST-CR-93-2 unless otherwise stated.

5.4.7.1.1 Antenna Location Record. The location of a radio transmitter's antenna shall be represented using an Antenna Location Record. This record shall specify the location of the radiating portion of the antenna. The antenna's location is specified in two different coordinate systems -- world coordinates and entity coordinates. The fields of this record are described in the paragraphs that follow.

(1) Antenna Location - This field shall specify the location of the radiating portion of the antenna. This field shall be represented by a World Coordinates Record (see 5.3.21).

(2) Relative Antenna Location - This field shall specify the location of the radiating portion of the antenna. This field shall be represented by an Entity Coordinate Vector Record (see 5.3.20.1)

The Antenna Location Record is represented in Figure 5-47.

Antenna Location	X-Coordinate - 64-bit float Y-Coordinate - 64-bit float Z-Coordinate - 64-bit float
Relative Antenna Location	X-Coordinate - 32-bit float Y-Coordinate - 32-bit float Z-coordinate - 32-bit float

Fig 5-47
Antenna Location Record

5.4.7.1.2 Antenna Pattern Parameter Record. The antenna Pattern Parameter Record shall be represented by one of the following records. Which record is used is determine by the value of the Antenna Pattern Type field. Antenna Pattern Types 0 and 1 shall be supported by all receiver simulators. Support for other Antenna Pattern Types is optional.

5.4.7.1.2.1 Omni-Directional Antenna Pattern Record. The Antenna Pattern Parameters field shall be an Omni-Directional Antenna Pattern Record when the Antenna Pattern Type field has a value of 0 (zero). This record indicates an antenna emitting uniformly in all directions. This pseudo-record contains no data and has a length of 0 (zero).

5.4.7.1.2.2 Beam Antenna Pattern Record. The Antenna Pattern Parameters field shall be Beam Antenna Pattern when the Antenna Pattern Type field has a value of 1. This record specifies the direction, pattern, and polarization of radiation from a radio transmitter's antenna. The radiation pattern produced by the antenna is defined with respect to a Beam Coordinate system. The Beam Coordinate system is a right-handed Cartesian coordinate system with the center of the beam along the x-axis of the system (See Figure 5-49). The Beam Antenna Pattern Record contains the following fields:

(1) Beam Direction - The beam direction field shall specify the rotation that transforms the reference coordinate system into the Beam Coordinate system. Either World Coordinates (see 5.3.21) or Entity Coordinates (see 3.2(11)) may be used as the reference coordinate system. The reference coordinate system in use shall be specified by the Reference System field of the Antenna Pattern Record. The beam direction field shall be represented as an Euler Angles Record (see 5.3.11).

(2) Azimuth Beamwidth - The.full width of the beam to the -3 dB power density points in the x-y plane of the Beam Coordinate system. Azimuth Beamwidth shall be represented by a 32-bit floating point number in units of radians.

(3) Elevation Beamwidth - The.full width of the beam to the -3 dB power density points in the x-z plane of the Beam Coordinate system. Elevation Beamwidth shall be represented by a 32-bit floating point number in units of radians.

(4) Reference System - Specifies the reference coordinate system with respect to which beam direction is specified. Reference System shall be represented as an 8-bit enumeration. An enumeration value of 1 shall represent World Coordinates. An enumeration value of 2 shall represent Entity Coordinates. No other values shall be used in this field. The value of this field shall not change over the duration an exercise. Different radios in the same exercise may use different reference systems. The World Coordinate System is the preferred reference system for training applications. The Entity Coordinate system should only be used where highly directional antennas must be precisely modeled. Use of the Entity Coordinates for a reference system requires the receipt of Entity State PDUs by radio simulators and may adversely affect their performance.

Support for an Entity Coordinate reference system is an optional part of this standard.

(5) E_z - The magnitude of the Z component (in beam coordinate) of the electrical field. E_z shall be represented by a 32-bit floating point number.

(6) E_x - The magnitude of the X component (in beam coordinate) of the electrical field. E_x shall be represented by a 32-bit floating point number.

(7) Phase - The phase angle between E_z and E_x in radians.

A fully omni-directional antenna shall be represented by Beam Direction Euler angles of zero, an Azimuth Beamwidth of 2 pi and an Elevation Beamwidth of pi.

The Beam Antenna Pattern Record is represented in Figure 5-48.

Beam Direction	Psi - 32-bit floating point Theta - 32-bit floating point Phi - 32-bit floating point
Azimuth Beamwidth	32-bit floating point
Elevation Beamwidth	32-bit floating point
Reference System	8-bit enumeration
Padding	24 bits unused
E_z	32-bit floating point
E_x	32-bit floating point
Phase	32-bit floating point

Fig 5-48
Beam Antenna Pattern Record

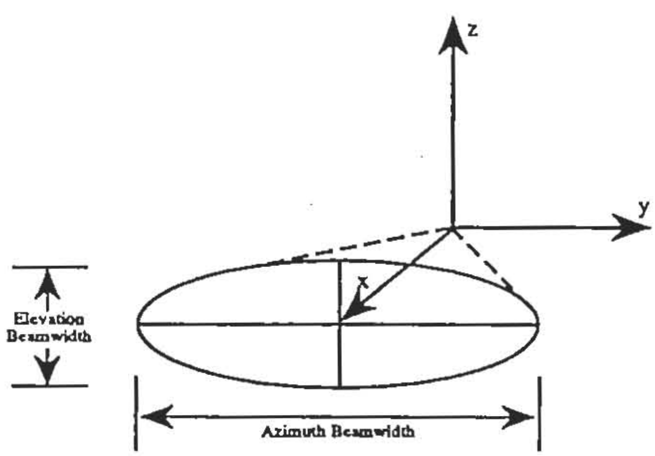


Figure 5-49
Beam Coordinates

5.4.7.1.2.3 Spherical Harmonic Antenna Pattern Record. The Antenna Pattern Parameters field shall be Spherical Harmonic Antenna Pattern Record when the Antenna Pattern Type field has a value of 2. This record specifies the direction and radiation pattern from a radio transmitter's antenna. The radio radiation pattern produced by the antenna may be represented either with respect to the World Coordinate System or the Entity Coordinate System. This antenna pattern does not support specifying polarization. The Spherical Harmonic Antenna Pattern Record contains the following fields:

(1) Order - Specifies the order of the expansion in spherical harmonics, counting from 0 (zero). There are 2n+1 coefficients for each order n, and a total of N^2+2N+1 coefficients through order N. Order shall be represented by an 8-bit unsigned integer.

(2) Coefficients - The coefficients field shall represent the power distribution from the antenna as the coefficients of a spherical harmonic expansion to the order given in the preceding field. Each coefficient shall be represented as a 32-bit floating point number. All cosine coefficients for a given order shall be given before all sine coefficients of that order. Equations for the expansion of a real function in spherical harmonics are in section 8 of document IST-CR-93-XX.

(3) Reference System - Specifies the reference coordinate system with respect to which beam direction is specified. Reference System shall be represented by an 8-bit enumeration. An enumeration value of 1 shall represent World Coordinates. An enumeration value of 2 shall represent Entity Coordinates. No other values shall be used in this field. The value of this field shall not change over the duration of an exercise. Difference radios in the same exercise may use different reference systems. The World Coordinate System is the preferred reference system for training applications. The Entity Coordinate system should only be used where highly directional antennas must be precisely modeled. Use of the Entity Coordinates for a reference system requires the receipt of Entity State PDUs by radio simulators and may adversely affect their performance. Support for an Entity Coordinate reference system is an optional part of this standard.

The Spherical Harmonic Antenna Pattern Record is represented in Figure 5-50.

Order	8-bit unsigned integer
Coefficients	Varies
Reference System	8-bit enumeration

Fig 5-50
Spherical Harmonic Antenna Pattern Record

5.4.7.1.3 Modulation Type Record. Information about the type of modulation used for radio transmission shall be represented by a Modulation Type Record. This record uniquely identifies the various sets of signal parameters (i.e. the modulation type) that are used to determine whether two radios may interoperate. The modulation is characterized in a generic fashion by the spread spectrum, major modulation type, and detail fields. The classes of interoperable modulation types are enumerated by the System field. This record shall specify the spread-spectrum usage, major modulation type, detailed information, and system compatibility. The fields for this record are described in the paragraphs that follow.

(1) Spread Spectrum - This field shall indicate the spread spectrum technique or combination of spread spectrum techniques in use. The Spread Spectrum field shall consist of a 16 element boolean array. Each independent type of spread spectrum technique shall be represented by a single element of this array. If a particular spread spectrum technique is in use, the corresponding array element shall be set to one, otherwise it shall be set to zero. All unused array elements shall be set to zero. The supported spread spectrum techniques and their assignment to elements of the 16 element array are defined in Section 4 of document IST-CR-93-2 and illustrated in Figure 5-51:

Bits 3-15	Bit 2	Bit 1	Bit 0
TBD	Time Hop	Pseudo Noise	Freq. Hop

Fig 5-51
Spread Spectrum Field Definition

(2) Major Modulation Type - This field shall specify the major classification of the modulation type. This field shall be represented by a 16-bit enumeration.

(3) Detail - This field shall provide certain detailed information depending upon the Major Modulation Type. This field shall be represented by a 16-bit enumeration.

(4) System - This field shall specify the interpretation of the modulation parameter field(s) in the Transmitter PDU. This field shall be represented by a 16-bit enumeration.

The Modulation Type record is represented on Figure 5-52.

Spread Spectrum	16-bit Boolean array
Major Modulation type	16-bit enumeration
Detail	16-bit enumeration
System	16-bit enumeration

Fig 5-52
Modulation Type Record

5.4.7.2 Transmitter PDU. Detailed information about a radio transmitter shall be communicated by issuing a Transmitter PDU. The Transmitter PDU shall contain the following fields:

(1) PDU Header - This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header Record described in 5.3.15.

(2) Entity ID - This field shall identify the entity that is the source of the radio transmission. The source entity may either represent the radio itself or represent an entity (such as a vehicle) which contains the radio. This field shall be represented by an Entity Identifier Record (see 5.3.8.2).

(3) Radio ID - This field shall identify a particular radio within a given entity. Radio IDs shall be assigned sequentially to the radios within an entity, starting with Radio ID 1. The combination of Entity ID and Radio ID uniquely identify a radio within a simulation exercise. The Radio ID field shall be represented by a 16-bit unsigned integer.

(4) Radio Entity Type - This field shall indicate the type of radio being simulated. This field shall be represented by a Radio Entity Type Record (see Section 6 of document IST-CR-93-2).

(5) Transmit State - This field shall specify whether a radio is off, powered but not transmitting, or powered and transmitting. This field shall be represented by an 8-bit enumeration.

(6) Input Source - This field shall specify which position (pilot, co-pilot, first officer, gunnery officer, etc.) or data port in the entity utilizing the radio is providing the input audio or data being transmitted. This

field shall be represented by an 8-bit enumeration. The values of this enumeration shall be independently specified for each Entity Type.

(7) Antenna Location - This field shall specify the location of the radiating portion of the antenna. This field shall be represented by an Antenna Location Record (see 5.4.7.1.1).

(8) Antenna Pattern Type - This field shall specify the type of representation used for the radiation pattern from the antenna. The value of this field shall determine the interpretation of the Antenna Pattern Parameter field. This field shall be represented by a 16-bit enumeration.

(9) Antenna Pattern Length - This field shall specify the length in octets of the Antenna Pattern Parameters field. This field shall be represented by an 16-bit unsigned integer.

(10) Frequency - This field shall specify the center frequency being used by the radio for transmission. This frequency shall be expressed in units of Hertz (Hz). This field shall be represented by a 64-bit unsigned integer.

(11) Transmit Frequency Bandwidth - This field shall identify the bandpass of the radio defined by the Radio ID field and the Radio Type field. This field shall be represented by a 32-bit floating point number.

(12) Power - This field shall specify the average power being transmitted in units of decibel-milliwatts (dBm). This field shall be represented by a 32-bit floating point number.

(13) Modulation Type - This field shall specify the type of modulation used for radio transmission. The modulation type shall be represented by a Modulation Type Record (see 5.4.7.1.3).

(14) Crypto System - This field shall identify the crypto equipment utilized if such equipment is used with the Transmitter PDU. This field shall be represented by a 16-bit enumeration.

(15) Crypto Key ID - This field shall consist of sixteen (16) bits. The high-order bit, when cleared, shall indicate that the crypto equipment is in the baseband encryption mode, and when set shall indicate that the crypto equipment is in the diphase encryption mode. The lower-order 15 bits shall be interpreted as a key identifier. If the key identifier of the transmitter and receiver

match, they shall be considered to be using the same encryption key. Note that this key identifier is not an actual crypto key, the nature of which varies among encryption systems and is classified.

(16) Length of Modulation Parameters - This field shall specify the length in octets of the modulation parameters that follow this field. This length shall include any end padding necessary to assure that the modulation Parameter field ends on a 64-bit boundary. Consequently, the value of this field shall be a multiple of 8. This field shall be represented by an 8-bit unsigned integer.

(17) Modulation Parameters - These fields shall specify specific modulation parameters for a radio transmission. This field shall have certain fields depending upon the value in the modulation type record. These fields shall be represented by Modulation Parameters fields.

(18) Antenna Pattern Parameters - These fields shall specify the radiation pattern from the antenna, its orientation in space, and the polarization of the radiation. These fields shall be represented by an Antenna Pattern Parameter Record.

The Transmitter PDU is represented in Figure 5-53.

FIELD SIZE (bits)	TRANSMITTER PDU FIELDS	
96	PDU HEADER	Protocol Version - 8-bit enumeration
		Exercise ID - 8-bit unsigned integer
		PDU Type - 8-bit enumeration
		Padding - 8 bits unused
		Time Stamp - 32-bit unsigned integer
		Length - 16-bit unsigned integer
		Padding - 16 bits unused
48	ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
16	RADIO ID	16-bit unsigned integer
64	RADIO ENTITY TYPE	Entity Kind - 8-bit enumeration
		Domain - 8-bit enumeration
		Country - 16-bit enumeration
		Category - 8-bit enumeration
		Subcategory - 8-bit enumeration
		Specific - 8-bit enumeration
		Extra - 8-bit enumeration
8	TRANSMIT STATE	8-bit enumeration
8	INPUT SOURCE	8-bit enumeration
16	PADDING	16 bits unused
192	ANTENNA LOCATION	X- Coordinate - 64-bit floating point
		Y- Coordinate - 64-bit floating point
		Z- Coordinate - 64-bit floating point
96	RELATIVE ANTENNA LOCATION	X- Coordinate - 32-bit floating point
		Y- Coordinate - 32-bit floating point
		Z- Coordinate - 32-bit floating point
16	ANTENNA PATTERN TYPE	16-bit enumeration
16	ANTENNA PATTERN LENGTH	16-bit unsigned integer

0049-2762.2

Figure 5-53
Transmitter PDU

FIELD SIZE (bits)	TRANSMITTER PDU FIELDS Cont'd	
64	FREQUENCY	64-bit unsigned integer
32	TRANSMIT FREQUENCY BANDWIDTH	32-bit floating point
32	POWER	32-bit floating point
64	MODULATION TYPE	Spread Spectrum - 16-Boolean Array
		Major - 16-bit enumeration
		Detail - 16-bit enumeration
		System - 16-bit enumeration
16	CRYPTO SYSTEM	16-bit enumeration
16	CRYPTO KEY ID	16-bit unsigned integer
8	LENGTH OF MODULATION PARAMETERS	8-bit unsigned integer
24	PADDING	24 bits unused
	MODULATION PARAMETER #1	
	⋮	
	MODULATION PARAMETER #N	
	ANTENNA PATTERN PARAMETER #1	
	⋮	
	ANTENNA PATTERN PARAMETER #M	

0049-2762.3

Figure 5-53 (Continued)
Transmitter PDU

5.4.7.3 Signal PDU. The actual transmission of voice or radio data shall be communicated by issuing a Signal PDU. The Signal PDU shall contain the following fields:

(1) PDU Header - This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header Record described in 5.3.15.

(2) Entity ID - This field shall identify the entity that is the source of the radio transmission. The source entity may either represent the radio itself or represent an entity (such as a vehicle) which contains the radio. This field shall be represented by an Entity Identifier Record (see 5.3.8.2).

(3) Radio ID - This field shall identify a particular radio within a given entity. This field shall be represented by a 16-bit unsigned integer.

The combination of Entity ID and Radio ID uniquely identifies a particular radio within a simulation exercise. The Entity ID, Radio ID pair associates each Signal PDU with the preceding Transmitter PDU that contains the same Entity ID, Radio ID pair.

(4) Encoding Scheme - This field shall specify the encoding used in the Data field of this PDU. The Encoding Scheme shall be represented by a 2-bit enumeration field indicating encoding class (voice, raw data, application-specific data, or pointer) and a 14-bit enumeration field representing the encoding type within that class (see Figure 5-54).

Bits 14-15	Bits 0-13
Encoding Class	Encoding Type

Fig 5-54
Encoding Scheme

The two most significant bits shall represent the following encoding classes:

00_2 = encoded voice
01_2 = raw binary data
10_2 = application-specific data
11_2 = pre-recorded voice pointer

The Data field specified in 5.4.7.3 (9). of this standard shall be interpreted differently depending on the encoding class identified by the upper two bits of the encoding scheme.

For encoded voice (Encoding Class 00_2), the Data field shall be interpreted as containing audio information digitally- encoded in accordance with the scheme identified by the Encoding Type (the least significant fourteen (14) bits of the Encoding Scheme field).

For raw binary data (Encoding Class 01_2), the Data field shall be interpreted as containing unspecified binary data. The Encoding Type has no significance in this case and shall be zero.

For application-specific data (Encoding Class 10_2), the Data field shall begin with a 32-bit user protocol identification number (data bytes 0-3), which identifies the application-specific encoding scheme or protocol in use. This Encoding Class shall be used to allow the development of various user defined synthetic protocols for communication among simulators (e.g. MCS, CVCC, etc.). The Encoding Type has no significance in this case and shall be zero.

For pointers (Encoding Class 11_2), the Data field shall be interpreted as a 32-bit pointer (data bytes 0-3) to the predefined database of messages, a 32-bit timestamp (data bytes 4-7) of when the message started being communicated, and a 32-bit duration of the message (bytes 8-11) in milliseconds. The Encoding Type has no significance in this case and shall be zero.

(5) Sample Rate - This field shall specify either (1) the sample rate in samples per second for voice digital audio or (2) the baud rate for data transmissions. This field shall be represented by a 32-bit unsigned integer. If the Encoding Class is "Pointer" (11_2), this field shall be zero.

(6) Data Length - This field shall specify the number of bits of digital voice audio or digital data being sent in this Signal PDU. This field shall be represented by a 16-bit unsigned integer. If the Encoding Class is "Pointer" (11_2), the Data Length field shall contain the value 96.

(7) Samples - This field shall specify the number of samples in this PDU. This field shall be represented by a 16- bit

unsigned integer. If the Encoding Class is not "Encoded Voice" (00_2), this field shall be zero.

(8) Data - This field shall specify the audio or digital data conveyed by the radio transmission. The length of the valid data contained in this field shall be the value of the Data Length field. This field shall be zero-padded to a length that is a multiple of 8 bits

The Signal PDU is represented on Figure 5-55.

FIELD SIZE (bits)	SIGNAL PDU FIELDS	
96	PDU HEADER	Protocol Version - 8-bit enumeration
		Exercise ID - 8-bit unsigned integer
		PDU Type - 8-bit enumeration
		Padding - 8 bits unused
		Time Stamp - 32-bit unsigned integer
		Length - 16-bit unsigned integer
		Padding - 16 bits unused
48	ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
16	RADIO ID	16-bit unsigned integer
16	ENCODING SCHEME	16-bit enumeration
16	PADDING	16 bits unused
32	SAMPLE RATE	32-bit integer
16	LENGTH	16-bit integer
16	SAMPLES	16-bit integer
8	DATA #0	8-bit unsigned integer
	⋮	
8	DATA # N	8-bit unsigned integer

0049-2763

Total Signal PDU Size = 256 + Length

Figure 5-55
Signal PDU

5.4.7.4 Receiver PDU. Communication of receiver state shall be communicated with a Receiver PDU. The Receiver PDU shall contain the following fields:

(1) PDU Header - This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header described in 5.3.15.

(2) Entity ID - This field shall identify the entity that is the source of the radio transmission. The source entity may either represent the radio itself or represent an entity (such as a vehicle) which contains the radio. This field shall be represented by an Entity Identifier Record (see 5.3.8.2).

(3) Radio ID - This field shall identify a particular radio within a given entity. Radio IDs shall be assigned sequentially to the radios within an entity, starting with Radio ID 1. The combination of Entity ID and Radio ID uniquely identify a radio within a simulation exercise. The Radio ID field shall be represented by a 16-bit unsigned integer.

(4) Receiver State - This field shall indicate the state of the receiver, which shall either be idle or active. This field shall be represented by a 16-bit enumeration.

(5) Received Power - This field shall indicate the RF power received, after applying any propagation loss and antenna gain. This field shall be represented by a 32-bit floating point number in units of dBm.

(6) Transmitter - This field shall identify the transmitter currently being received. The selection of the received transmitter depends on the of the characteristics and state of the simulated receiver. It shall consist of an entity ID and a radio ID, as specified in paragraphs 2 and 3, above.

The Receiver PDU is represented on Figure 5-56.

INSERT FIGURE 5-56. RECEIVER PDU

FIELD SIZE (bits)	RECEIVER PDU FIELDS	
96	PDU HEADER	Protocol Version - 8-bit enumeration
		Exercise ID - 8-bit unsigned integer
		PDU Type - 8-bit enumeration
		Padding - 8 bits unused
		Time Stamp - 32-bit unsigned integer
		Length - 16-bit unsigned integer
		Padding - 16 bits unused
48	ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
16	RADIO ID	16-bit unsigned integer
16	RECEIVER STATE	16-bit enumeration
16	PADDING	16-bits unused
32	RECEIVED POWER	32-bit floating point
48	TRANSMITTER ENTITY ID	Site - 16-bit unsigned integer
		Application - 16-bit unsigned integer
		Entity - 16-bit unsigned integer
16	TRANSMITTER RADIO ID	16-bit unsigned integer

0049-2983

Total Receiver PDU Size = 288 bits

Figure 5-56
Receiver PDU

This page intentionally left blank.

6. Experimental Protocols

6.1 Introduction. This sections contains specifications for experimental protocols for DIS. The content of this section is intended for experimental purposes. This section is not mandatory.

6.1.1 Aggregate Protocol. The following paragraphs shall establish the content of the Aggregate Protocol PDUs and the procedure for using the Aggregate Protocol in a DIS exercise.

6.1.1.1 Purpose. The Aggregate Protocol provides a mechanism for grouping multiple entities and communicating information about these groups of entities. It also provides a mechanism for participants in a DIS exercise to request that the entities that make up an aggregate join or leave the DIS exercise.

6.1.1.1.1 Scope. This standard provides a mechanism for SAFORs and constructive wargames to interface to DIS exercises by describing aggregates. The coordination of the simulation of constituent entities, the assignment of de-aggregated entities to simulation applications is outside the scope of this standard.

6.1.1.1.2 Requirement. The Aggregate Protocol is an optional part of the DIS Protocol. Participants in a DIS exercise do not have to process Aggregate Protocol PDUs beyond what is necessary to ignore the Aggregate Protocol PDUs.

6.1.1.1.3 Origination. The Aggregate Protocol PDUs may be transmitted and received by a Simulation Management application other than the application simulating its constituent entities. The Simulation Management application that issues and receives Aggregate PDUs is called the Aggregate Controller. Exercises may have multiple Aggregate Controllers. The Aggregate Controller may be at a different Site than its constituent entities.

6.1.1.1.4 Aggregate Controller. The Aggregate Controller is responsible for issuing in and receiving Aggregate Protocol PDUs.

6.1.1.1.5 Timing. The Aggregate PDUs are issued every 5 seconds, or at the default interval that Entity State PDUs are issued.

6.1.1.2 Aggregate Definition. An aggregate is a group of entities . Entities may be grouped by type, capabilities, mission, command hierarchy, or arbitrarily. The purpose of an aggregation is to save bandwidth by sending fewer PDUs for the Aggregate than for individually represented units, or to save computational effort by providing information on the group of entities that can be used by a receiver to sort Entity State PDUs, or to put hierarchical information about military or functional units or groups on the DIS network.

6.1.1.2.1 Issuance of the aggregate protocol PDUs. If the entities that make up an Aggregate are in compliance with the DIS standard, they will always issue Entity State PDUs. In practice however, many uses of the Aggregate Protocol, such as interfaces to SAFORs or Constructive Wargames, will only issue the Entity State PDUs for the subset of the Aggregates that are currently participating in DIS exercises.

6.1.1.2.2 Joining a DIS exercise. The process of joining a DIS exercise via the Aggregate Protocol is called de-aggregation. It implies that the constituent entities of an aggregate were not issuing or receiving DIS PDUs and after de-aggregation are fully compliant with the DIS protocols. The aggregate descriptor PDUs are issued before and after de-aggregation.

6.1.1.2.3 Leaving a DIS excise. The process of leaving a DIS exercise via the Aggregate Protocol is called re-aggregation. It implies that the constituent entities of an aggregate were issuing or receiving DIS PDUs and after re-aggregation will not issue DIS PDUs. The aggregate descriptor PDUs are issued before and after re-aggregation.

6.1.1.3 Aggregate Protocol PDUs. The Aggregate Protocol defines three PDUs and also uses the Acknowledge PDU of the Simulation Management Protocol. The Four PDUs in the Aggregate Protocol are:

(1) Aggregate Descriptor PDU
(2) De-Aggregate Request PDU
(3) Re-Aggregate Request PDU
(4) Acknowledge PDU (see section 4.4.6.4.5)

6.1.1.3.1 PDU Header. All Aggregate Protocol PDUs contain the standard PDU header (see 4.4.1).

6.1.1.4 Aggregate Descriptor PDU. The Aggregate Descriptor PDU consists of a header, and zero or more Aggregate Descriptors fields.

6.1.1.4.1 Aggregate Descriptor Header. The number of Aggregate Descriptor fields is in the Aggregate Descriptor PDU in addition to the standard PDU header information.

6.1.1.4.2 Aggregate Descriptor Field. Each Aggregate Descriptor field describes the location, orientation and shape of one Aggregate. An aggregate Descriptor field contains either a list of the Unique Entity IDs of the Entities that make up the aggregate, or a list of the Unique Aggregate ID of its subordinate Aggregates. The Aggregate Descriptor field should only list Entity IDs for entities that are issuing Entity State PDUs.

An Aggregate Descriptor Field with zero (0) entities or subordinate aggregates means the aggregate is on the lowest hierarchical level and is not issuing DIS PDUs (has not been de-aggregated).

Entities can move from one aggregate to another if their entity IDs are listed in a different descriptor field. Entities ID may only be listed in one Aggregate Descriptor at a time. Aggregate can change hierarchy by changing the descriptor field. Multiple level aggregates should always describe all levels of the Aggregate, even if one or more of the lower level aggregates have been de-aggregated.

6.1.1.4.2.1 Unique Aggregate ID. The unique aggregate ID is assigned by the Aggregate Controller. It is unique in the same manner entity IDs are unique. Aggregate IDs may be a duplicate of Entity IDs.

6.1.1.4.2.1 Number of Entities in Aggregate. Number of entities or subordinate entities in aggregate field. An aggregate can contain either constituent entities or subordinate aggregates, but not both. This means that either M or m must equal zero. A non-zero value in one of those fields represents the number of entities or subordinate aggregates within the Aggregate Descriptor. Figure 3 represents an Aggregate Descriptor containing entities (M>0). Figure 4 represents an Aggregate Descriptor of containing subordinate aggregates (m>0).

6.1.1.4.2.2 Aggregate Shape Field. This field describes the shape or formation of the entities within the aggregate. The formations are aligned on the orientation vector. The first bit describes the whether or not the entities are on the earth's surface. Bits 1 to 15 describe the shape of the aggregate. Figure 7 describes the contents of bits 1 to 15.

6.1.1.4.2.3 Aggregate Flag. Aggregate Flag is an enumerated type that delineates the type of aggregate Figure 6 represents the values of the Aggregate Flag.

Unique Aggregate ID	16-bit unsigned integer
Size of Descriptor (in bytes)	16-bit unsigned integer
Aggregate Shape	16-bit enumeration
Number of Entities in Aggregate (M)	16-bit unsigned integer
Number of Subordinate Aggregates (m)	16-bit unsigned integer
Aggregate Hierarchical Name	16-bit enumeration
Length of Rectangular Solid Enclosing Aggregate Along X Velocity Axis	32-bit floating point
Length of Rectangular Solid Enclosing Aggregate Along Y Velocity Axis	32-bit floating point
Length of Rectangular Solid Enclosing Aggregate Along Z Velocity Axis	32-bit floating point
Position of Aggregate Center	X: 64-bit floating point Y: 64-bit floating point Z: 64-bit floating point
Aggregate Orientation	Psi: 32-bit floating point Theta: 32-bit floating point Phi: 32-bit floating point
Aggregate Velocity	X: 32-bit floating point Y: 32-bit floating point Z: 32-bit floating point
Unique Entity ID #1	16-bit Site ID 16-bit Application ID 16-bit Entity ID
. .	. .
Unique Entity ID # M	16-bit Site ID 16-bit Application ID 16-bit Entity ID

Figure 6-1
Entity Aggregate Descriptor Record

Unique Aggregate ID	16-bit unsigned integer
Size of Descriptor (in bytes)	16-bit unsigned integer
Aggregate Shape	16-bit enumeration
Number of Entities in Aggregate (M)	16-bit unsigned integer
Number of Subordinate Aggregates (m)	16-bit unsigned integer
Aggregate Hierarchical Name	16-bit enumeration
Length of Rectangular Solid Enclosing Aggregate Along X Velocity Axis	32-bit floating point
Length of Rectangular Solid Enclosing Aggregate Along Y Velocity Axis	32-bit floating point
Length of Rectangular Solid Enclosing Aggregate Along Z Velocity Axis	32-bit floating point
Position of Aggregate Center	X: 64-bit floating point Y: 64-bit floating point Z: 64-bit floating point
Aggregate Orientation	Psi: 32-bit floating point Theta: 32-bit floating point Phi: 32-bit floating point
Aggregate Velocity	X: 32-bit floating point Y: 32-bit floating point Z: 32-bit floating point
Unique Aggregate ID #1	16-bit unsigned integer
. .	. .
Unique Aggregate ID # m	16-bit unsigned integer

Figure 6-2
Subordinate Aggregate Descriptor Record

PDU Header	PDU Header Record
Originating Entity ID	Site: 16-bit unsigned integer Application: 16-bit unsigned integer Entity: 16-bit unsigned integer
Number of Aggregate Descriptors (N)	32-bit unsigned integer

Figure 6-3
Aggregate Descriptor PDU

6.1.1.5 Aggregation and De-Aggregation Protocol. Any entity can request to an Aggregate Controller that it de-aggregate by sending a De-Aggregate Request PDU. The Aggregate Controller receives the request and returns a De-Aggregate Acknowledge PDU. The constituent entities of the Aggregate then issue their Entity State PDUs.

6.1.1.5.1 Aggregate Controllers can request to re-aggregate. The aggregate sends out a warning to all other participants that it is going to re-aggregate by issuing a Re-Aggregate Request PDU. Any entity can stop this process by sending a De-Aggregate Request PDU to the Aggregate Controller.

6.1.1.5.2 Issuance of De-Aggregate Request. In order to prevent "flooding" of the network by an ill-timed Re-Aggregate request, entities should send out De-Aggregate Requests at a random time 0 to 5 seconds after receipt of an undesired Re-Aggregate Request PDU. Entities may listen for other entities sending out De-Aggregate requests in order not to duplicate the request.

6.1.1.5.3 Receipt of De-Aggregate Request. Upon receipt of the De-Aggregate Request the receiving Aggregate Controller communicates to its component entities that they should now issue Entity State PDUs. The required mechanism for this is outside the scope of this standard. The Simulation Management Protocol (4.4.6) may be used for this role.

6.1.1.5.4 Detection Range. The Detection range of the sending entity can be used by the Aggregate to help determine when it can re-aggregate. If a detection range value received by an Aggregate Controller is greater than the default detection range of the Aggregate Controller's re-aggregate function, then the new detection range can be used for re-aggregation calculations.

The responsibility for de-aggregating when an aggregate comes within interaction range of DIS entities always rests with the aggregate controller. The detection range is provided only as an aid. The de-aggregation protocol is not meant imply that DIS entities are responsible for "cueing" aggregate controllers when to de-aggregate, since the entire aggregate protocol is optional.

PDU Header	PDU Header Record
Originating Entity ID	Site: 16-bit unsigned integer Application: 16-bit unsigned integer Entity: 16-bit unsigned integer
Unique Aggregate ID of Receiving Aggregate Descriptor	16-bit unsigned integer
Padding	32 bits unused
Detection Range of Sending Entity	64-bit floating point

Figure 6-4
De-Aggregate Request PDU

6.1.1.6 Re-Aggregate Request PDU. This PDU is sent by the Aggregate Controller before it re-aggregates and stops sending Entity State PDUs for its component entities.

6.1.1.6.1 Re-Aggregation Timing. The time of re-aggregation must be at least 10 seconds after the time the PDU is issued. This is to enable a receiving entity or an exercise controller to stop the re-aggregation. This is done by sending a De-Aggregate Request PDU in the manner described above.

PDU Header	PDU Header Record
Originating Entity ID	Site: 16-bit unsigned integer Application: 16-bit unsigned integer Entity: 16-bit unsigned integer
Unique Aggregate ID of Sending Aggregate Descriptor	16-bit unsigned integer
Simulation Time When Re-Aggregate Will Occur	Hours: 32-bit integer Time Past The Hour: 32-bit unsigned integer

Figure 6-5
Re-Aggregate Request PDU

6.1.1.7 Description of the De-Aggregation Process. This will describe the process of taking an aggregated unit from an aggregated state to a de-aggregated state. There are multiple paths through the protocol from an aggregated state to a de-aggregated state.

6.1.1.7.1 The unit starts aggregated. It may chose at any time to become de-aggregated. This can be accomplished without any interaction with other entities or management entities on the network.

6.1.1.7.2 The unit starts aggregated. The Aggregate Controller receives a de-aggregate request. It responds with an acknowledge PDU with the acknowledge flag set to "will comply". It then enters a de-aggregated state.

6.1.1.7.3 The unit starts aggregates. The Aggregate Controller receives a de-aggregate request. It responds with an acknowledge PDU with the acknowledge flag set to "will not comply". It remains aggregated.

6.1.1.8 Description of the Re-Aggregation Process. This will describe the process of taking an aggregated unit from a de-aggregated state to the aggregated state. There is one path from the De-Aggregated State to the Aggregated state but, it can be interrupted.

6.1.1.8.1 The unit starts de-aggregated. The Aggregate Controller starts the re-aggregation process by issuing a Re-Aggregate Request PDU. It is now waiting to re-aggregate.

6.1.1.8.2 Aggregate Timing. After ten seconds in the Waiting to Re-aggregate State the unit enters the Aggregated State unless it receives and responds positively to a De-Aggregate Request.

Insert figure 6-6 Aggregate protocol state diagram

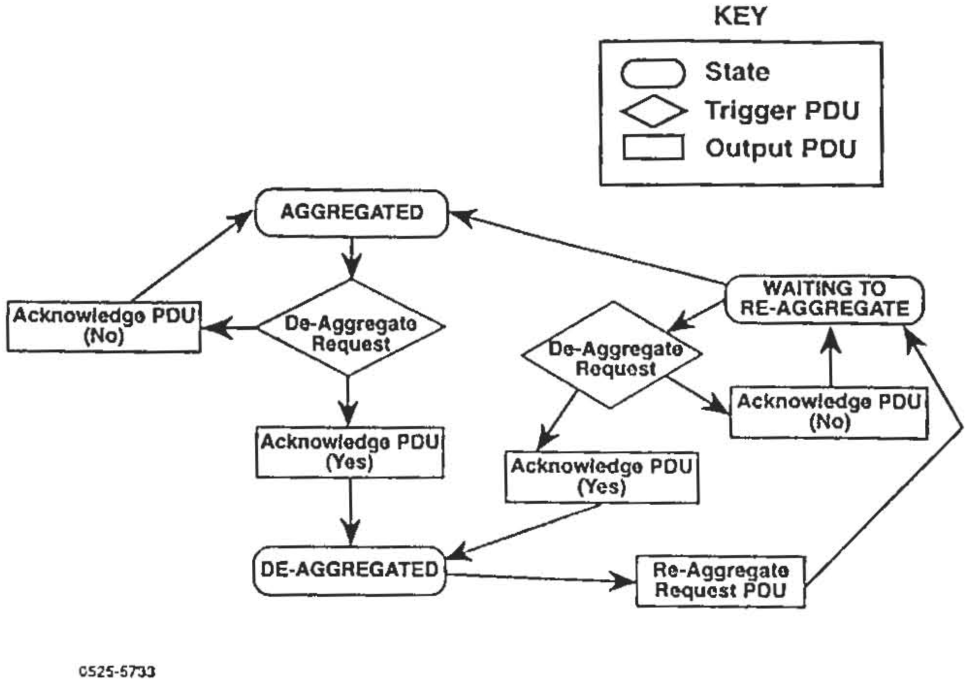


Figure 6-6

Aggregate Protocol State Diagram

Enumeration Tables for Aggregate Protocol

Value	Description
0	undefined or unknown type of group
1	single
2	squad
3	platoon
4	company
5	battalion
6	regiment
7	division
8	task force
9	squadron

Table 6-1
Aggregate Flag

Value	Description
0	No specific Shape
1	Line
2	Column
3	Double Column
4	Triple Column
5	Vee (or Vic)
6	Inverted V (or Vic)
7	Wedge
8	Inverted Wedge
9	Circle (or Laager)
10	Diamond
11	Herringbone
12	Evenly Spaced Within Solid Rectangle
13	Evenly Spaced Within Largest Solid Spheroid Defined

Table 6-2
Bits 1-15 of the Aggregate Shape Field

Value	Description
bit 0 = 0	Entity Conform to the terrain. (lay flat on the Earth)
bit 0 = 1	Entity do not conform to the earth's surface

Table 6-3
Bit 0 (zero) of the Aggregate Shape Field

ANNEX A - Articulated and Attached Parts

A1. Scope

A1.1 Scope. This annex defines the mechanism by which the state of articulated and attached parts shall be communicated. The particular parts that are to be represented as articulated rather than fixed are not specified by this standard. This annex is a mandatory part of the standard.

A2. Applicable Documents

The following document is referenced in this Annex:

IST-CR-93-02 - Enumeration and Bit Encoded Values for Use with Protocols for Distributed Interactive Simulation Applications Version 2.0 (Second Draft)

A3. Articulated Parts and Attached Parts

A3.1 Introduction. The Articulation Parameter Record can be used in two ways. The first is to represent articulated parts of an entity. The second is to represent attached parts. The representation of Articulated Parts is described in A3.2 and Section 5 in document IST-CR-93-02. The representation of Attached Parts is described in A3.3 and Section 5 in document IST-CR-93-02.

A3.2 Articulated Parts. The Articulation Parameter Record is used to represent the state of the movable parts of an entity. Examples of movable parts include the turret on a tank and the periscope on a submarine. An Articulation Parameter Record shall represent the value of only one parameter of a movable or "articulated" part. Thus, it may require multiple articulation parameter records to describe the state of a single articulated part. The number of Articulation Parameter Records used to represent a given articulated part on a given entity shall be determined at exercise initialization. It shall not change for the duration of the simulation exercise. The parameter type field of the Articulation Parameter Record shall identify both a particular movable part and the type metric (state parameter) whose value is contained in the parameter value field (see A3.2.3). The 16 currently defined type metrics (state parameters) are described under A3.2.4.

A3.2.1 Numbering of Articulated Parts. Each of an entity's articulated parts shall be sequentially assigned a part ID. The entity itself shall be assigned part ID 0 (zero). An articulated part shall have a part ID greater than that of the articulated part to which it is attached. The part IDs associated with an entity shall be a continuous sequence starting with zero. The part ID is used in the "ID - Part Attached To" field of the Articulation Parameter Record (see 5.3.3).

The part ID associated with an Articulated Parts Record is expressed by its order within an Entity State or Detonation PDU. The first Articulated Part Record in a PDU is associated with part ID 1. For subsequent Articulated Part Records, the associated part ID is incremented whenever the type class (see A3.2.5) in the record changes.

A3.2.2 Reference and Part Coordinate Systems. Each articulated part has an associated part coordinate system. This coordinate system is fixed with respect to the part. Each articulated part also has an associated reference coordinate system. This coordinate system is fixed with respect to the part that the articulated part is attached to. Both the part and reference coordinate system are right-handed Cartesian coordinate systems. Both coordinate systems are defined when the entity type is defined.

The collection of Parameter Types and Parameter Values fields for a part in a PDU define the transformation from the reference coordinate system to the part coordinate system. In general, the reference and part coordinate systems should be defined such that a null transformation corresponds to the part in its "neutral" position. Additional guidelines for defining these coordinate systems are given in Section 5 in document IST-CR-93-02.

A3.2.3 Parameter Type Field. An articulated parameter type consists of three components (Fig A-1). The first component, consisting of the low 5 bits of the type, defines the type metric. The type metric determines which of the transformations described in A3.2.4 are specified by this parameter type. The second component, consisting of the next 26 bits of the type, defines the type class. The type class specifies which particular articulating part of a model is referenced by this parameter type. The values of the parameter metrics are defined in Section 5 in document IST-CR-93-02. The most significant bit of the Parameter Type Record is set to 0 to indicate that the Articulation Parameter Record represents an Articulated Part.

```
31  30                      5 4                 0
0  ¦  <- Type Class ->       ¦  <- Type Metric ->¦
```

Fig A-1
Parameter Type Field Format - Articulated Part

A3.2.4 Type Metrics

A3.2.4.1 Position. Position shall specify the location of an articulated part along a particular path to which its movement is constrained. The path may be any 3-space curve. The position shall be expressed in 32-bit floating point numbers. The value zero shall represent fully retracted, and one shall represent fully extended. Intermediate positions are represented as a fraction of the path traveled. *One path may be associated with each articulated part on each entity type.*

A3.2.4.2 Position Rate. Position rate shall specify the rate of change of position in units of fraction of entire path per second. For example, a position rate of one indicates that the articulated part has traversed the entire path in one second. Position rate shall be expressed in 32-bit floating point format.

A3.2.4.3 Extension. Extension shall specify the linear extension of the part of one direction in meters. The value zero shall represent fully retracted. Extension shall be expressed in 32-bit floating point format.

A3.2.4.4 Extension Rate. Extension rate shall specify the rate of change of extension in units of meters per second. The extension rate shall be expressed in 32-bit floating point format.

A3.2.4.5 x, y, and z. The x, y, and z shall specify the translation from the articulated parts reference coordinate system to the current location of the articulated parts coordinate system. x, y, and z shall be expressed in 32-bit floating point format.

A3.2.4.6 x, y, and z Rates. The x, y, and z rates shall specify the rate of change of the position of the articulated coordinate system expressed in meters per second. x, y, and z rates shall be expressed in 32-bit floating point format.

A3.2.4.7 Azimuth. Azimuth shall specify the rotation of an articulated part with respect to its reference z-axis. Azimuth shall be measured in Radians and shall be expressed in 32-bit floating point format.

A3.2.4.8 Elevation. Elevation shall specify the rotation of an articulated part with respect to its reference y-axis. Elevation shall be measured in Radians and shall be expressed in 32-bit floating point format.

A3.2.4.9 Rotation. Rotation shall specify the rotation of an articulated part with respect to its reference x-axis. Rotation shall be measured in Radians and shall be expressed in 32-bit floating point format.

A3.2.4.10 Order of Transformation. If more than one of azimuth, elevation, or rotation exist for the same articulated part, the order of the transformation shall be azimuth, then elevation, then rotation.

A3.2.4.11 Azimuth, Elevation, and Rotation Rates. These rates shall specify the rate at which the angle is changing. Angular rates shall be measured in Radians per second and shall be expressed in 32-bit floating point format. These rates represent instantaneous angular velocity.

A3.2.5 Type Class. The type class uniquely identifies a particular articulated part on a given entity type. Guidance for uniquely assigning type classes to an entity's articulated parts is given in Section 5 in document IST-CR-93-02.

A3.2.6 Dead Reckoning of Articulated Parts. The issuing simulation application may specify the dead reckoning of an articulated part by including position, extension, x, y, z, azimuth, elevation, and/or rotation rate type metrics in an Articulation Parameter Record of an Entity State PDU (see 4.4.2.1). The same dead reckoning thresholds (see 4.4.2.1.2.1) that apply to an entity apply to its articulated parts. Note that angular movement of an articulated part may result in a linear threshold being exceeded. Absolute linear error for each articulated part in a chain of connected parts shall be used to determine if a threshold has been exceeded.

A3.2.7 Guidelines for Selecting Type Metrics. Although it is possible to define part and reference coordinate systems so that almost any type metric could apply to almost any articulated part, Table A-1 provides a guideline for which type metrics should be used with some common articulated parts. Rate parameters shall be used when dead reckoning of articulated parts is required.

Table A-1
Guidelines for Selecting Type Metrics

Part	Recommended Type Metric
Horizontal control surfaces	Elevation
Vertical control surfaces	Azimuth
Extendable items	Extension
Fixed path items	Position
Turrets	Azimuth
Guns	Elevation
Movable missile launcher	Azimuth and elevation

Rate parameters shall be used when dead reckoning of articulated parts is required.

A3.2.8 Parameter Value Field. The 64-bit Parameter Value field is divided into two 32-bit sub-fields. The most-significant 32-bit sub-field represents a 32-bit floating point number. The interpretation of this sub-field depends on the value of type metric as specified in A3.2.4. The least significant 32-bit sub-field shall be zero.

A3.3 Attached Parts. The Articulation Parameter Record is also used to represent removable parts that may be "attached" to an entity. Examples of such "attached parts" include missiles and other external stores on an aircraft. When the Articulation Parameter Record is used in this way, the parameter type field shall identify the location (or station) to which the part is attached (see Fig A-2). Stations shall be assigned to each entity type as described in the subparagraph that follows. When the Articulation Parameter Record is used to represent an attached part, the parameter value field shall contain the entity type of the attached part. Entity types are defined in Section 6 in Document IST-CR-93-02. When the most significant bit of the Parameter Type Record is set, the Articulation Parameter Record represents an Attached Part.

31	30	0
1	Station	

Fig A-2
Parameter Type Field Format - Attached Part

A3.3.1 Stations. A station shall specify the station to which an attached part is attached. Stations shall be numbered sequentially beginning with one and incrementing by one. The order of numbering shall be from top to bottom, then back to front, then left to right. The only exception shall be aircraft wing stations. The fuselage stations, left wing stations, and right wing stations shall be separated into different categories. Wing stations shall be numbered from inboard to outboard. The value field shall contain the Entity Type Record representing the type of store located at the specified station. The attached part stations are defined in Section 5 in Document IST-CR-93-02.